QUANTIFICATION AND EXTRACTION OF SURFACE FEATURES FROM DIGITAL TERRAIN MODELS

by

Mikhail Minin, BSc.

Brock University

Submitted

in partial fulfillment of the requirements

for the degree of

Master of Science in Earth Sciences

Faculty of Mathematics and Sciences, Brock University

St. Catharines, Ontario

Abstract

Digital Terrain Models (DTMs) are important in geology and geomorphology, since elevation data contains a lot of information pertaining to geomorphological processes that influence the topography. The first derivative of topography is attitude; the second is curvature. GIS tools were developed for derivation of strike, dip, curvature and curvature orientation from Digital Elevation Models (DEMs). A method for displaying both strike and dip simultaneously as colour-coded visualization (AVA) was implemented. A plug-in for calculating strike and dip via Least Squares Regression was created first using VB.NET. Further research produced a more computationally efficient solution, convolution filtering, which was implemented as Python scripts. These scripts were also used for calculation of curvature and curvature orientation. The application of these tools was demonstrated by performing morphometric studies on datasets from Earth and Mars. The tools show promise, however more work is needed to explore their full potential and possible uses.

Acknowledgements

I would like to express deep gratitude to Frank Fueten for making this thesis possible. I wish to thank Bob Stesky for many rounds of revisions, Mariek Schmidt and Bill Morris for providing feedback, Rick Cheel for support and advice. I would like to express appreciation to all the faculty and staff at Brock University for understanding and cooperation. I also would like to thank all my friends and especially my mom.

Table of Contents

List of Abbreviations

AVA = Augmented Visualization of Attitude

DEM = Digital Elevation Model

DTM = Digital Surface Model

DTM = Digital Terrain Model

e.g. = *exempli gratia*, for example

HRSC = High Resolution Stereo Camera

HSL = Hue, Saturation, Lightness (colour representation)

i.e. = *isto es*, that is

MAE = Mean Absolute Error

MD = Maximum Deviation

MSE = Mean Squared Error

MSSD = Mean Signed Squared Differece

RGB = Red, Green, Blue (colour representation)

SSE = Sum of Squared Errors

List of Figures

Chapter 1: Overview

INTRODUCTION

Analysis of topography can provide insight onto the processes that formed it. Vast volumes of high-resolution elevation data are collected by satellites and airborne surveys. A software environment capable of processing geospatial information is referred to as Geographic Information System (GIS). The most commonly used such system is ESRI ArcGIS. ArcGIS contains Spatial Analyst extension designed specifically for analysis of topography (i.e., geomorphometry); however, ArcGIS has some limitations.

Contour maps, shaded relief and other common topographic representations do not show quantitative information regarding surface attitudes. Morphometric analysis requires knowledge of surface derivatives such as slope and aspect, i.e. the steepness of the tangent plane and the downslope direction [Behrendt, 2012]. In geology, the synonymous terms are the "dip" and the "dip direction"; however, the pair "strike and "dip" is more commonly used. The strike, in accordance with the "American right hand rule" [Davis, 2002], is the azimuth 90º counterclockwise (CCW) from the dip direction.

Morphometric information is important in physical geography, particularly for hydrological and solar insulation studies [Slocum et al., 2009]. For this, ArcGIS provides Aspect and Slope built-in tools in the Surface toolset, available in the ArcGIS Spatial Analyst® as well as in the ArcGIS 3D Analyst® extensions. These are incapable of estimating goodness-of-fit or of using a moving window of sizes larger than 3x3 pixels. Here, a moving window means a neighborhood set of pixels on a DEM, whose elevation data is used for approximating the surface derivative at the central pixel location [ArcGIS Desktop Help 10.0, Appx. 1].

Multiple third-party applications for visualisation and analysis of digital surfaces already exist, e.g. Jo Wood's LandSerf software [LandSerf Manual, Appx. 1]. LandSerf is a free package containing extensive tools and an intuitive scripting language. It possesses capabilities for multi-scale quadratic regression, yet lacks the support for geoTIFF format [LandSerf Manual, Appx. 1].

This thesis presents morphometric geoprocessing tools developed to be accessible from within the ArcMap environment and tailored for structural geology applications. Two approaches to the generation of DEM's first derivative have been developed: a command tool written with VB.NET in Visual Studio, and an ArcGIS toolbox containing Python scripts and geoprocessing models. The command tool for computation of surface first derivative (i.e., attitude of the surface) is discussed in Chapter 2. A more efficient method, implemented as Python script, is discussed in Chapter 3. Chapter 4 reviews application of Python scripts to computation of second derivative (i.e. curvature), as well as a general process for computing higher order derivatives. Chapter 5 demonstrates application of tools presented herein for morphometric analysis of three geological examples. The first example maps extensional and compressional features on Tharsis rise. The second example computes frequency distribution analysis of attitudes on spur-and-gully textures on the walls of Hebes Chasma, Valles Marineris, Mars. The spur-and-gully morphology is a pattern of repeated sharp rocky ridges known as referred to as spurs, and intervening depressions, called gullies [Hargitai, 2015]. The last example compares frequency distribution for curvature orientations between a sample region of a known horst and graben texture in Iceland and different chasmata of Valles Marineris, Mars.

BACKGROUND

Digital Elevation Model (DEM), Digital Surface Model (DSM) and Digital Terrain Model (DTM) are terms frequently used interchangeably, yet subtle differences do exist. DSM means a set of points in 3D space, representing vertices of an interpolated mesh assumed to be connected according to proximity, whereas each vertex is a vector with 3 degrees of freedom. This term generally refers to the first reflection obtained from a remote sensing device, usually a LiDAR point cloud [Behrendt, 2012]. DTM is a generic term for any projected digital representation of a surface relief [Heywood et al., 2006]. Since DSM is a mesh (a network of connected vertices in 3D space), and DTM is a projection (an image of a 3D object plotted onto a reference surface), DSM may represent surfaces with overhangs, which DTM does not allow. The term DEM, in its original sense, applied to any point height dataset, independent of the spacing of data points [Heywood et al., 2006]. However, modern usage refers exclusively to point height data on a regularly spaced grid (i.e., a raster representation) [Goodchild and Kemp, 1990]. In this work, consistent with commonly accepted use, the term DEM refers to a DTM raster displacement map [Sclaroff and Pentland, 1991] over locally flat datum (i.e., reference surface).

As with any raster, a DEM has a constant-interval grid spacing, with array indices corresponding to the position of elements in each of the n Cartesian dimensions and scaling that may vary between dimensions, but not within. Every picture element (pixel) in the DEM is assigned a scalar value corresponding to elevation above the datum at that pixel's centre, the geographic coordinates of which are determined by its array indices (i.e., the position within the raster). Due to limitations of tabular data storage, no pixel on a DEM can contain more than one value. Consequently, a DEM surface cannot be

overturned, meaning it cannot be flipped upside down or contain overhangs, such as caves [Minar and Evans, 2008]. Hence, a DEM is a representation of an n-dimensional continuously infinitely differentiable scalar field [Borisenko and Tarapov, 1966] for a function $F = (x, \dots)$. Nevertheless, a representation is never perfect, and derivatives are undefined at pixels bordering undefined regions, such as on the edges of the raster and near pixels with NODATA values.

Most DEMs are 2D raster representations of a planetary surface with elevation measured from an arbitrary datum, usually a geoid (e.g., sea level on Earth) then projected onto a graticule (i.e., geographic coordinate system) of a best-fit ellipsoid at the study location, followed by a cartographic re-projection onto a plane. [Slocum et al., 2009]

The more exotic applications of DEMs include pollution tracing [Heywood et al., 2006], watershed segmentation for grain boundary identification in petrographic studies [Barraud, 2006] and 3D plots from magnetic resonance imaging [Johnson et al, 2015, ITK software guide, Appx. 1]. Although the methods discussed herein are of interdisciplinary value, exotic uses fall beyond the scope of this research.

While some of the techniques presented are scalable to higher dimensions [Russ, 1992], our tools were designed to process only 2D rasters since in planetary crustal studies a geologist working with topography only needs to be concerned with physical 2D surfaces in real, 3D space.

A common technique to produce a DEM of a planetary surface is by triangulation from remotely sensed stereo imagery, as described by Gonzalez and Woods [1992]. DEMs studied in this thesis were produced this way using the NASA AMES stereo pipeline [Moratto et. al., 2010; Broxton and Edwards, 2008], unless stated otherwise.

A derivative of a two-dimensional scalar field is a vector field with two degrees of freedom. These vectors can be represented as plane normal directional cosines (i.e., gradients). The gradients can be converted into the values of aspect and slope, or strike and dip.

While the dip values can be clearly displayed using a monochrome colour ramp, strike/aspect requires a different approach. As azimuths of 0º and 360º are equivalent, so must be identical the colours representing these values in an effective visualization; yet, they must still uniquely display the values in-between. That can be accomplished by using the gradient of different hues.

Consequently, implementations of colour wheel visualizations of directional data have been developed as part of this thesis, and these representations were named Augmented Visualization of Attitude (AVA). The term AVA was invented to describe colour-coded attitude visualization, where surfaces are coloured to represent the orientation of the normal axis. Since the impetus for this work was initially to aid in structural geological analysis, the term AVA was chosen so as to reflect similarity to Achsensverteilungsanalysen (axial-distribution analysis), A.V.A. developed by Bruno Sander for tectonite fabric analysis [Turner and Weiss, 1963]

Using colour saturation to represent dip values gives AVA an appearance similar to hillshading. However, while hillshading is unidirectional, the AVA highlights irregularities in all directions, providing a clear view of subtle features. The AVA display of surface attitudes is much clearer and more detailed than Aspect, Slope or Hillshade.

Similar approaches to attitude visualizations have been explored in the past. An effective method for visualization of Aspect has been developed by Moellering and Kimerling [1990], who first noticed that since colours of different hues with same

lightness and saturation values have different luminance, encoding aspect as hue on a hue-saturation-lightness (HSL) model distorts perception of attitude differences.

Subsequently, Brewer and Marlow [1993] incorporated slope into their visualization as saturation, while compensating for perceptual bias due to variable relative luminance by rectifying HSL model onto a topological manifold of Munsells hue-value-chroma (HVC) colour space [Slocum et al., 2009] (Fig. 1)

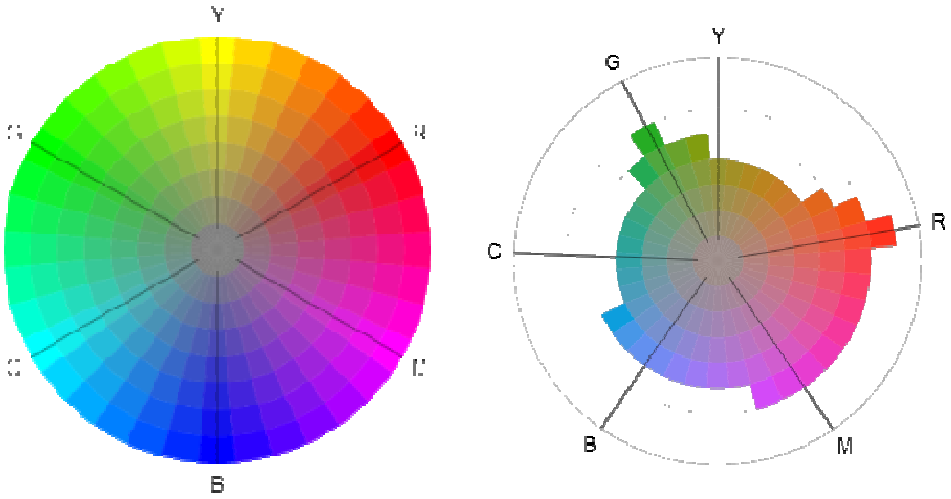


Figure 1: Discrepancy between computer graphics colour display and human eye colour perception becomes apparent when comparing HSL colour wheels with (right) and without (left) compensations for relative luminance [http://www.redskyworld.com/biggrumpything/?p=413]

Work by Brewer and Marlow [1993] categorized attitudes into eight classes of aspect and three classes of slope, represented by hue and saturation respectively. The add-on described in the next chapter increments values of hue or saturation with every degree of change in either. Not compensating for relative luminance bias simplifies data conversion at a cost of perceptual distortion [Slocum et al., 2009]. An option to use custom colour maps was introduced in the script version of the tool, described in Chapter 3.

Since determining the tangent plane attitude involves fitting a plane to a group of pixels, the goodness of fit can be measured by comparing the predicted and actual altitude for every pixel within a neighbourhood.

In this thesis, three approaches to the generation of the DEM's gradients have been developed, namely *Least Squares Regression*, *Nearest Neighbour Differences*, and *Convolution Filtering*. The subsequent processing steps differ only in implementation; hence, the overall process flow for computing first derivative is conceptually the same for all methods (Fig. 2):

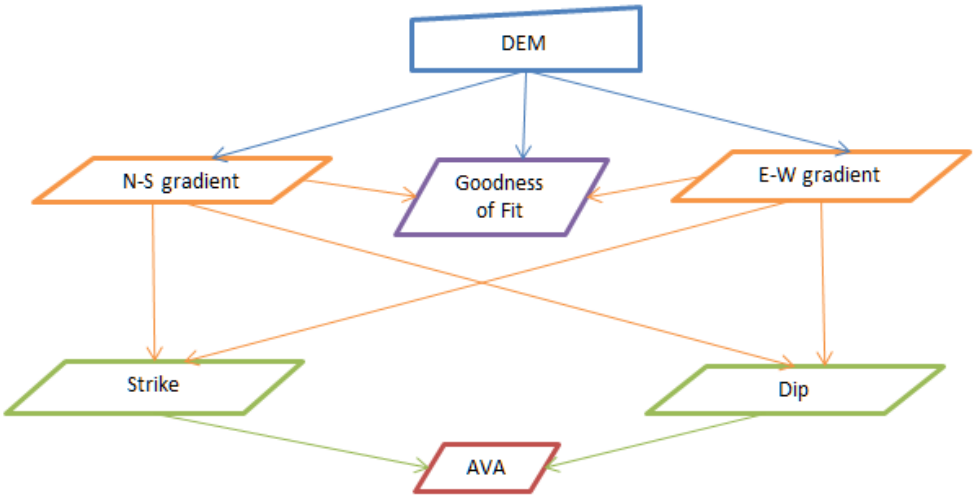


Figure 2: Conceptual diagram describing the process flow

The process begins with computation of gradients from the DEM. The *"Goodness of Fit"* can be calculated based on the differences between the actual values on the DEM and the values predicted by gradients. The "Strike" and "Dip" is calculated from gradients trigonometrically. The AVA is produced by assigning a colour to each pixel as dictated by its Strike and Dip values; this is achieved by means of a look-up table.

Derivative tools were designed in an abstract XYZ-coordinate space. For working with planetary data, this thesis matches X-axis with East-West direction and Y-axis with South-North direction.

Chapter 2: A Command Tool implementing Least Squares Regression for the quantification of topographic features

INTRODUCTION

Common methods for calculating from DEM the orientation of tangent planes have been reviewed by Skidmore [1989]:

1. Second-order finite difference method on a 2x2 moving window [Dozier and Strahler, 1983];
2. Third-order finite difference method with [Horn, 1981] or without [Sharpnack and Akin, 1969] a weighting factor for non-diagonally adjacent cells on a 3x3 moving window;
3. Multiple linear regression models where a surface is fitted to a 3x3 neighbourhood, by calculating regression coefficients using co-variance matrices [Neter and Wasserman, 1974].

Most research has not considered using moving windows larger than 3x3 pixels.

To assist geologists who conduct morphometric analysis on a planetary surface, an ArcGIS Command Tool add-on has been created (Fig. 3).

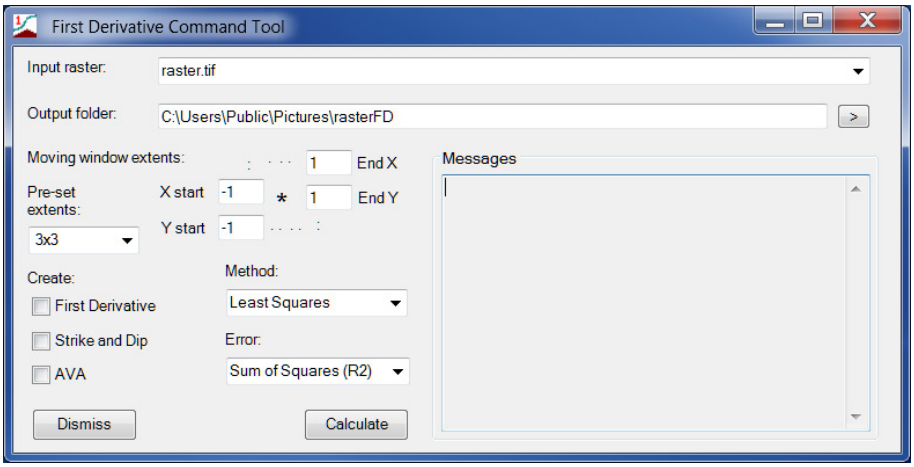


Figure 3: Dialog box for the in-house-developed First Derivative Tool

It is capable of processing GeoTIFFs, using other-than-Earth coordinate systems, implements custom size sampling windows, estimates goodness of fit and produces a comprehensive visualization of the DEM first derivative.

METHODS

Programming environment and data management

This add-on was written in VB.NET using Microsoft Visual Studio 2010 integrated development environment, and assembled as a Dynamically Linked Class Library (.dll). Source code consists of two files: *DerivativeSelectedLayerCmd.vb* contains the code necessary to register the add-on as a command component within ArcGIS; *Dialog.vb* contains all the code for displaying the user interface and running this add-on.

Clicking the command button for this add-on evokes the dialog box shown in Figure 3. When the user clicks on *Calculate* button, add-on performs data validation, and, if entered parameters appear to be correct, it proceeds to create raster datasets on the hard drive; else, it displays an error message.

This tool writes all outputs to the *Output folder*, whose default name is derived from the name of the input raster without extension, appended with letters "FD". If desired, a different folder may be chosen by the user, yet all outputs to this folder are named automatically. This tool has the capacity to produce up to three output files:

1. ***The First Derivative*** file contains parameters to the equation of the best-fit plane. It uses the name of the source file appended with letters "_FD.tif", this raster has 4 bands, each as 32 bit floating point number. This file uses 16 bytes per pixel.
2. ***The Strike and Dip*** file contains three channels: the Strike, Dip and Error. Its filename is appended with letters "_SD.tif". The channels in this raster are also 32 bit floating point numbers, hence this file uses 12 bytes per pixel
3. ***The AVA*** file is colour-coded attitude visualization, appended with "_AVA.tif". This is a three-channel, RGB raster, containing 8 bits per channel, 3 bytes per pixel.

Since pixel data is not directly editable, access to it is gained via *IPixelBlock* iterator interface. The maximum size of the pixel block is limited by the available memory. Consequently, it is not always possible to use the pixel block large enough to fit the entire image. Furthermore, writing to TIFF datasets in blocks of sizes other than the default 128x128 px TIFF image tiles was discovered to corrupt the output.

Since derivatives are undefined for pixel block boundaries, the input block sizes must exceed the sizes of output blocks by the half-span of a sampling window. When the add-on is run, the total number of blocks and the number of block processed is displayed in the *Messages* text field. Once calculations complete the add-on requests whether to add the outputs to the map and, if so, it imports them as new layers to the ArcGIS Table of Contents. Clicking *Dismiss* closes the dialog.

Computing Cartesian plane equation parameters

Two approaches for calculating the gradients were implemented in this add-on. The first method involves standard Least Squares Regression. The second method, referred to as the Mean of Nearest Neighbour Differences, approximates a gradient as average of nearest neighbour differences. Both methods employ square moving windows of sampling regions. In this work the size of the sampling region is given as either length by width, or as a half-span, i.e. the distance, in pixels, from the centre of the window to the edge, e.g. a window 3x3 px has a half-span of 1x1 px.

Least Squares Regression

Least Squares Regression (LSR) is a technique for approximating polynomial equation by minimizing the Sum of Squares of differences between observed and predicted values for a set of data points [Davis, 2002].

Similar to a method effectively implemented in Orion™ software [R. Stesky, Appx. 1], *N* data points $P(x_i, y_i, z_i);\ i = 1, \ldots, N;$ are used to fit a model of a plane, where $ax + by + cz + d = 0$ is the equation of a plane; $\bar{n} = (a, b, c)$ is the vector of the plane normal; *x*, *y* are model arguments, and *z* is the dependent variable.

On a continuously differentiable scalar field, such as a DEM, vertical slopes are not allowed. Consequently, the vertical component of tangent plane normal, *c*, can never be zero. Dividing through by *c*, ratios *a/c*, *b/c* and *d/c* may be replaced by new coefficients *a'*, *b'*, and *d'*, so that $z = -(a'x + b'y + d')$. The

$$Sum\ of\ Squares = \sum_{i=0}^{N} [z_i - z(x_i, y_i; a', b', d')]^2 = \sum_{i=0}^{N} [z_i - a'x_i + b'y_i + d']^2$$

is minimized when its partial derivatives with respect to a', b', and d' are all zero. Differentiation returns a system of equations [Appx. 1], in matrix notation written as

$$\begin{bmatrix} \sum_{i=1}^{N} x_i^2 & \sum_{i=1}^{N} y_i x_i & \sum_{i=1}^{N} x_i \\ \sum_{i=1}^{N} x_i y_i & \sum_{i=1}^{N} y_i^2 & \sum_{i=1}^{N} y_i \\ \sum_{i=1}^{N} x_i & \sum_{i=1}^{N} y_i & N \end{bmatrix} \begin{pmatrix} a' \\ b' \\ d' \end{pmatrix} - \begin{bmatrix} -\sum_{i=1}^{N} z_i x_i \\ -\sum_{i=1}^{N} z_i y_i \\ -\sum_{i=1}^{N} z_i \end{bmatrix};$$

This matrix equation can be solved by Gaussian elimination [Roberts, 1982; Boyce and DiPrima, 2009] for any set of more than three non-coplanar points.

Mean of Nearest Neighbour Differences

This method calculates average gradient along columns and rows. Only the elevations on opposite sides of a moving window are compared, since the differences between adjacent pixels cancel out upon summation [Appx. 5.1]. Hence, since the run against which rise is measured is one pixel less than the span of the sampling window,

$$\Delta X = \frac{1}{M(N-1)}\left(\sum_{j=1}^{M} x_{1,j} - \sum_{j=1}^{M} x_{N,j}\right); \quad \Delta Y = \frac{1}{(M-1)N}\left(\sum_{i=1}^{N} y_{i,1} - \sum_{i=1}^{N} x_{i,M}\right).$$

where ΔX is a gradient along the X-axis (i.e., East-West) and ΔY $\boldsymbol{\Delta Y}$ is a gradient along the Y-axis (i.e., North-South), while *N* and *M* are the spans (i.e., length and width) of the moving window. Evidently, for $z_{x,y} = -(a'x + b'y + d')$, $a' = -\Delta X$ and $b' = -\Delta Y$.

The value of the coefficient d' $\boldsymbol{d'}$ represents the elevation at the origin, i.e. value of z when x and y are both zero. It is computed from gradient as: $d' = z_{x,y} - a'x - b'y$.

Mean of Nearest Neighbor Differences is not equivalent to Least Squares Regression. It is less accurate, but requires fewer calculations, and therefore is slightly faster. The methods were compared in the chapter's discussion.

Estimating goodness of fit

In addition to a choice of two methods for calculating the attitude of the surface, four methods for calculating goodness of fit of the tangent plane are provided in this add-

on. All methods (Figs 7 and 8) operate by measuring the dispersion of sample points relative to the model plane, with control variables x and y and the test variable z:

1. ***Mean Squared Error (MSE)***

$Error = \frac{1}{N}\sum(True\ value - Predicted\ value)^2$; taken over all N pixels on the moving window. Since the predicted value is $z'_{x,y} = -(a'x + b'y + d')$, and true value is $z_{x,y}$;
$MSE = \frac{1}{N}\sum\left(z_{x,y} - z'_{x,y}\right)^2 = \frac{1}{N}\sum\left(z_{x,y} + a'x + b'y + d'\right)^2$. (Fig. 7.a)

2. ***Mean Absolute Error (MAE)***

$MAE = \frac{1}{N}\sum\left|z_{x,y} - z'_{x,y}\right| = \frac{1}{N}\sum\left|z_{x,y} + a'x + b'y + d'\right|$. (Fig. 7.b)

3. ***Maximum Deviation (MD)***

$MD = Maximum\left(\left|z_{x,y} - z'_{x,y}\right|\right) = Maximum\left(\left|z_{x,y} + a'x + b'y + d'\right|\right)$ (Fig. 7.c)

4. ***Mean Signed Squared Difference (MSSD)***

$$MSSD = \frac{1}{N}\sum\left(\left(z_{x,y} - z'_{x,y}\right)\left|z_{x,y} - z'_{x,y}\right|\right) = \frac{1}{N}\sum\left(\left(z_{x,y} + a'x + b'y + d'\right)\left|z_{x,y} + a'x + b'y + d'\right|\right).$$ (Fig. 8)

While the first three techniques effectively highlight irregular areas, they fail to distinguish between convex and concave curvatures. The fourth technique overcomes this issue via multiplying deviations by their absolute values prior to averaging.

Calculating Strike and Dip

Using the Cartesian plane equation parameters, the strike and dip of the plane can be calculated. The vector normal to a plane is $\vec{n} = (a, b, c) = (a', b', 1)$ and its length is $L = \sqrt[2]{a'^2 + b'^2 + 1}$; hence, the unit normal vector is $\hat{n} = \frac{1}{L}(a', b', 1) = \left(\frac{a'}{L}, \frac{b'}{L}, \frac{1}{L}\right)$.

With this, the trend and plunge of the plane normal can be calculated [Appx. 5.3.1, 5.3.2]:
$Trend = \tan^{-1}\left(\frac{\hat{n}_x}{\hat{n}_y}\right) = \tan^{-1}\left(\frac{a'/L}{b'/L}\right) = \tan^{-1}\left(\frac{a'}{b'}\right)$; $Plunge = \sin^{-1}(\hat{n}_z) = \sin^{-1}\left(\frac{1}{L}\right) = \sin^{-1}\left(\frac{1}{\sqrt[2]{a'^2 + b'^2 + 1}}\right)$.

Alternatively, the plunge may also be found by projecting the normal vector onto the horizontal plane:

$$n_{projXY} = \sqrt[2]{a'^2 + b'^2}; \quad Plunge = \tan^{-1}\frac{c'}{n_{projXY}} = \tan^{-1}\frac{1}{\sqrt[2]{a'^2 + b'^2}}.$$

The attitude of plane normal is easily convertible into the attitude of the plane itself:

$$Strike = Trend - 90 = -\tan^{-1}\frac{b'}{a'} = \tan^{-1}\frac{-b'}{a'}; \quad Dip = 90 - Plunge = \tan^{-1}\sqrt{a'^2 + b'^2}.$$

Since Strike and Dip are independent from coefficient a' d', attitude of the tangent plane is the same as the attitude of the plane of best fit.

Converting Strike and Dip to Augmented Visualization of Attitude

Strike and dip values were interpreted as HSL colour model coordinates, such that

Hue = Strike; *Saturation = Dip / 90;* *Lightness = 0.5*

HSL values were converted to standard RGB representation by an algorithm adapted from Agoston [2005]. No attempt was made at this stage towards colour corrections to compensate for relative luminance perceptual bias.

RESULTS

The tool was tested over a sample 50 m/px DEM representing a segment of the walls of Valles Marineris, Mars, that display spur-and-gully morphology [Hargitai, 2015; Patton, 1990; Peulvast et al., 2001 (Fig.4).

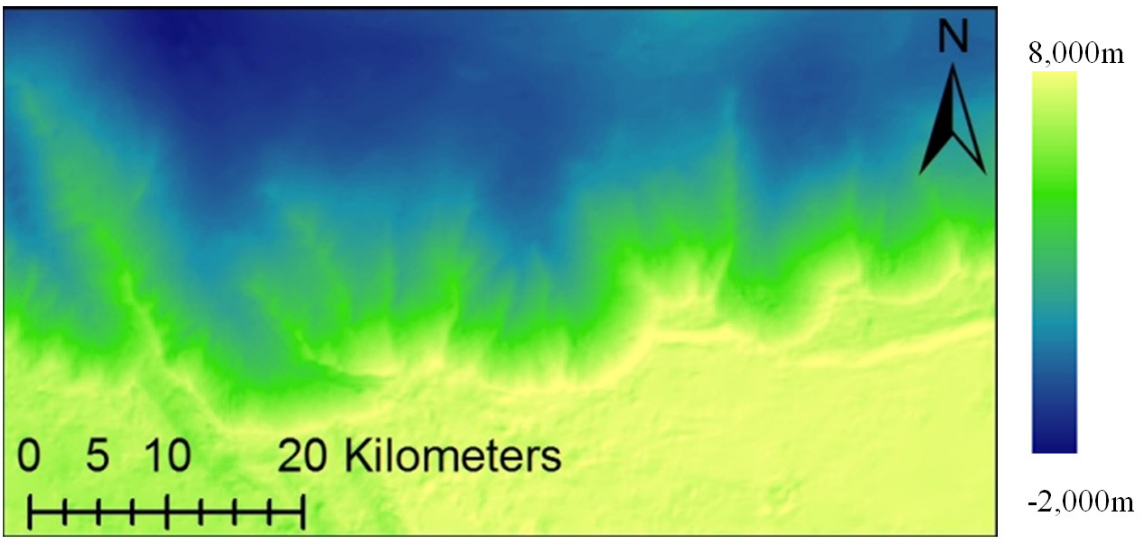


Figure 4: Sample DEM, 1516x875 px, displayed using the colour ramp on the right, with the ArcGIS hillshading effect (z = 0.2). For consistency, all images in this and the third chapter are of the same location, unless noted otherwise.

Intermediate outputs, the raster FD, containing the – plane equation parameters (Fig. 5.a), and raster SD, containing the – Strike, Dip and Error (Fig. 5.b) were computed.

These intermediate files are not meant for visual assessment; the data contained within is intended for numerical work, such as segmentation and statistics. The strike would always be in the range from 0° to 360°, and dip between 0° and 90º, and the error, gradient and offset would be between the minimum and the maximum for that image.

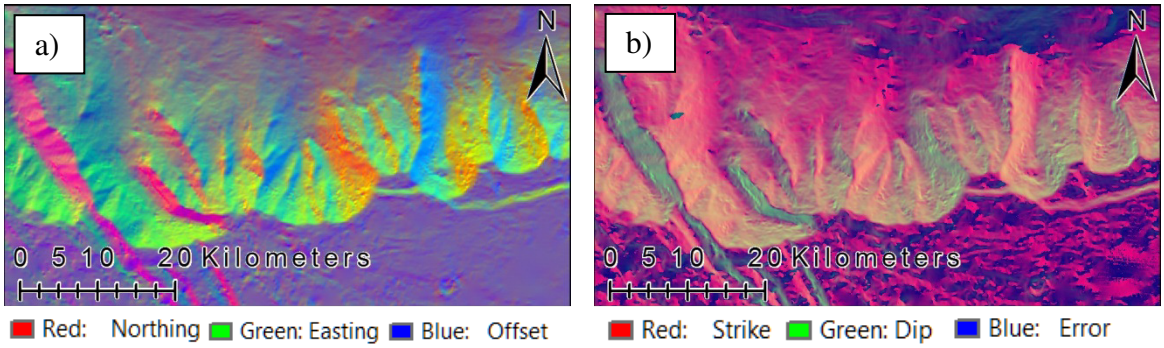


Figure 5: Outputs of the first derivative tool as 3 band rasters: a) FD raster, b) SD raster. Colour coding is equalized by histograms using Percent Clip method, with Gamma Stretch applied. Each channel on a raster represents a different property, as described by the legend. In this particular example, the ranges for FD raster are: -0.76 to 1.19 for Y-gradient (Northing); -0.98 to 1.05667 for X-gradient (Easting); -64,179.9 m to 41,144.6 m for offset.
For SD raster, 0 to 360 for Strike, 0 to 90 for Dip, and 0 to 62.38 for Error.

Sample outputs for goodness of-fit estimates are shown in Figures 6 and 7.

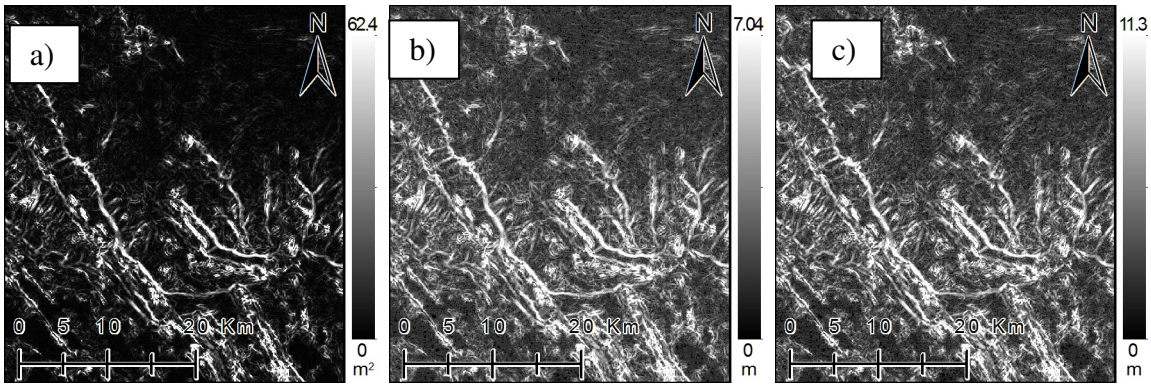


Figure 6: Different methods of error estimation supported by this add-on: a) Mean Squared Error, b) Mean Absolute Error, c) Maximum Deviation; all obtained with a 3x3 kernel and displayed using Standard Deviations stretch type at n =2.5

MSE provides the cleanest result, while Maximum Deviation the most detailed (Fig. 6). Mean Signed Squared Differences, in addition to measuring the extent of deviation, also allows distinguishing between concave and convex irregularities. (Fig. 7)

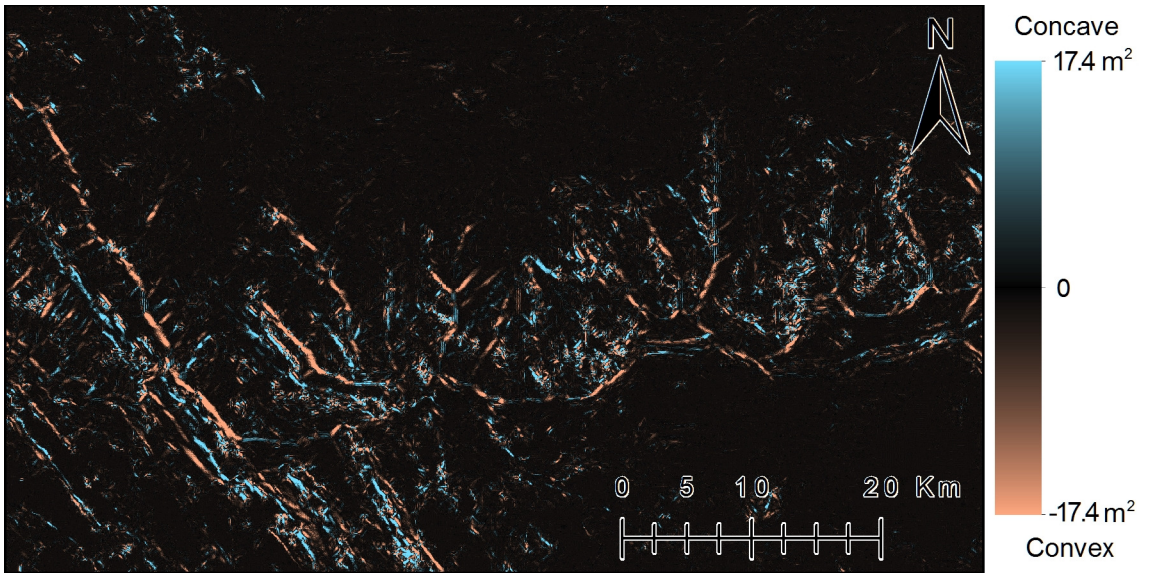


Figure 7: Mean Signed Squared Difference of a sample DEM, colour coded using a ramp displayed on the right.

AVA were produced using a 3x3 px moving window for the sample image as well as for a computer generated DTM of a concave surface of a half-sphere. (Fig. 8)

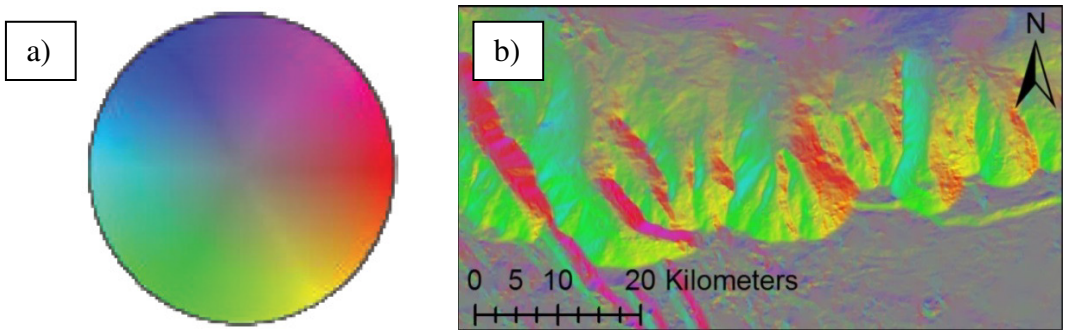


Figure 8: Augmented Visualization of Attitude (AVA) produced by Least Squares Regression with a square, 3x3 px moving window:
a) when run over perfect concave half-sphere DEM
b) when running over sample DEM

The accuracy of strike and dip outputs from Least Squares Regression were tested against the reference value obtained with *Orion*^{TM} software. The case study DTM had a

grid spacing of 4.88x4.88 m/px. The feature of interest was a flank of a spur [Hargitai, 2015; Patton, 1990; Peulvast et al., 2001], with width of 250 to 350 meters, i.e. 51 to 71 px. (Fig. 9)

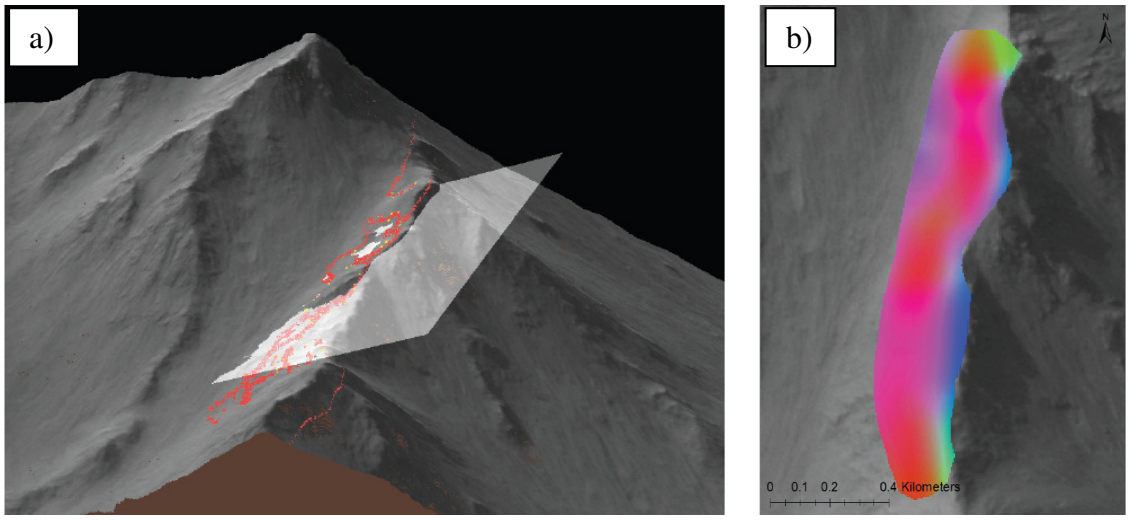


Figure 9: Flank of a spur planar topographic feature:
a) selected using Orion™ software [after Smith, 2013] and
b) extracted by polygon from LSR Command Tool outputs, colour wheel is as shown in figure 8

The attitude of the flank was measured in *Orion™* using a set of manually defined points on the DEM. The LSR tool was run over the same DEM with moving windows of different sizes. The flank of the spur region was extracted from the output rasters by a manually drafted polygon mask. The attitudes of individual pixels were exported as a *CSV ASCII* text file. From *ASCII* file, histograms were produced and plotted against Orion™ measurement by means of *Python 2.7* using the *matplotlib.pyplot* module and the *numpy* package. The frequency peaks for attitude were found to correlate well with the values obtained from Orion™. (Fig. 10)

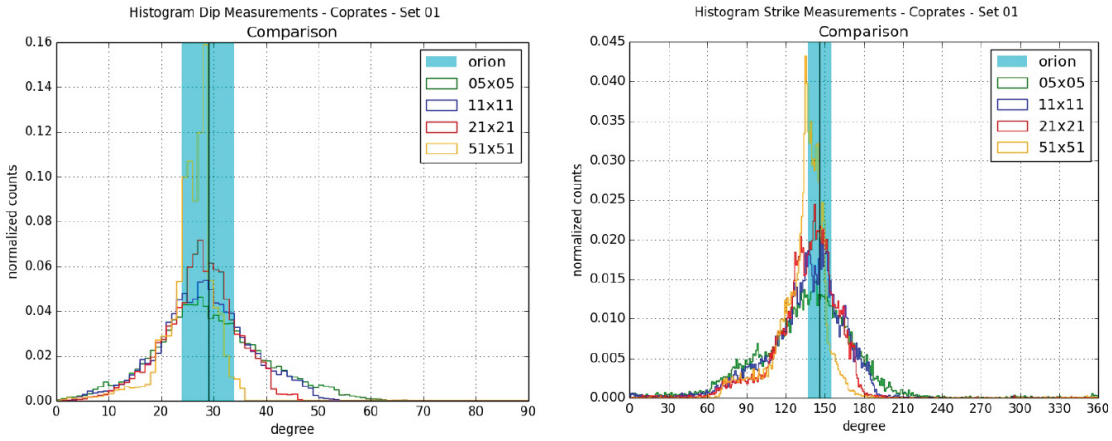


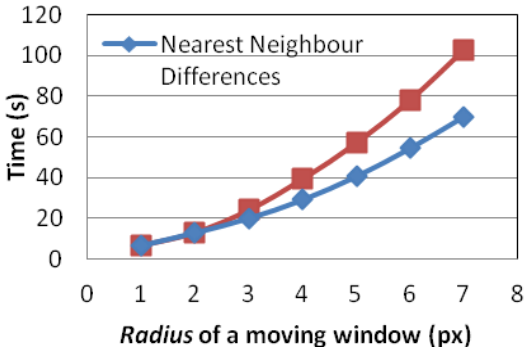
Figure 10: Histogram comparison [performed by Vargas, L. E., 2014] shows good correlation. The blue bar indicates confidence interval in terms of standard deviation from the best-fit plane modeled with Orion™.

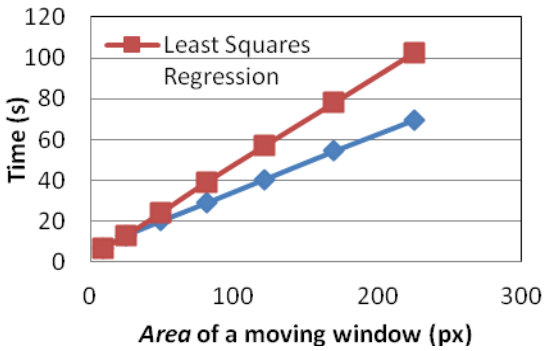
The correlation tend to improve with increase in size of the moving window, yet regression with sampling region of 51x51 px returned less accurate results than with 21x21 px. Precision, i.e. height of peaks in normalized counts, continued to increase due to greater smoothing. Evidently, expanding the sampling region improves the accuracy of the regression, but only up until its size becomes comparable with the size of the feature. Further increases in neighbourhood size result in model disregarding the feature as noise.

DISCUSSION

The tool developed herein allows for calculation of planar attitudes using a variety of different kernel sizes. Numerically the results agree well with the measurements made using the manual Orion technique. On rough or noisy surfaces, minor undulations obfuscate larger features. Using the existing ArcGIS surface tools on noisy DEMs requires pre-processing by low-pass filtering, i.e. averaging, whenever roughness is an issue. Averaging entails loss of information. In contrast, expanding the moving window mitigates errors due to high frequency noise, while also improving result accuracy. Data quantity boosts regression quality as long as the moving window remains sufficiently

small relative to the features of interest. However, the computational time increases with the size of DEM and with the moving window size. A 1514x873 px test image was processed using sampling windows of different sizes, and corresponding computational times presented in Figure 11. Computing times grow proportionally to the area of a moving window, i.e. with the square of the moving window half-span.





Moving window	Half-span	1x1	2x2	3x3	4x4	5x5	6x6	7x7
	Size	3x3	5x5	7x7	9x9	11x11	13x13	15x15
	Number of pixels	9	25	49	89	121	169	225
Computational time	Nearest Neighbour Differences	6.83s	13.07 s	20.10s	29.10s	40.58s	54.33s	1m 9.78s
	Least Squares Regression	6.83s	13.15 s	24.19s	39.31s	57.02s	1m 18.00s	1m 42.66s

Figure 11: Processing times of a sample image, 1514 by 873 px, compared between NND and LSR for various neighbourhood sizes.
Scatter plots (above) and data table (below).

To evaluate the discrepancy between the outputs of the Nearest Neighbour Differences and the Least Squares Regression, the values for Strike and Dip were obtained from a sample dataset using various sizes of sampling window. Outputs from the two methods were rounded to a nearest degree and their absolute differences calculated. (Fig. 12)

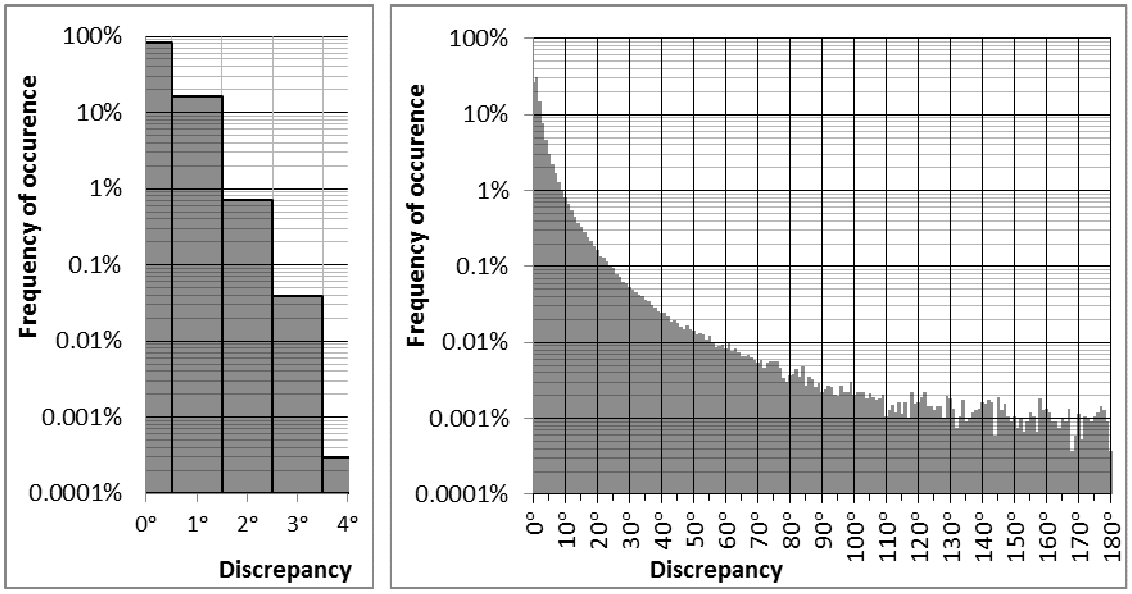


Figure 12: Histograms for comparison of LSR and NND outputs of Strike and Dip, when made with 15x15 square kernels.

For Strike, the discrepancy was most apparent near flat areas, while for dip values, variations were more obvious at slopes. The difference of as much as 12 degrees was recorded for Dip at a 3x3 moving window. For a 7x7 moving window differences between two methods became much less apparent for both Strike and Dip, with Dip values being no more than 6 degrees apart. Strike value anomalies became more frequent in the nearly horizontal regions. For moving window of 15x15 pixels, Strike anomalies became very prominent in the nearly horizontal regions, yet Dip anomalies nearly disappeared, with maximum difference of only 5 degrees. (Fig. 13)

Nearly horizontal planes have a small range of dip values. Hence, it is not surprising that the absolute difference between dip values is more at the slopes. Similarly, strike values become more prone to rounding differences in the two techniques as the surface dips less. This is clearly visible in the nearly horizontal plateau region in the southern portion of the sample image (Fig. 13e).

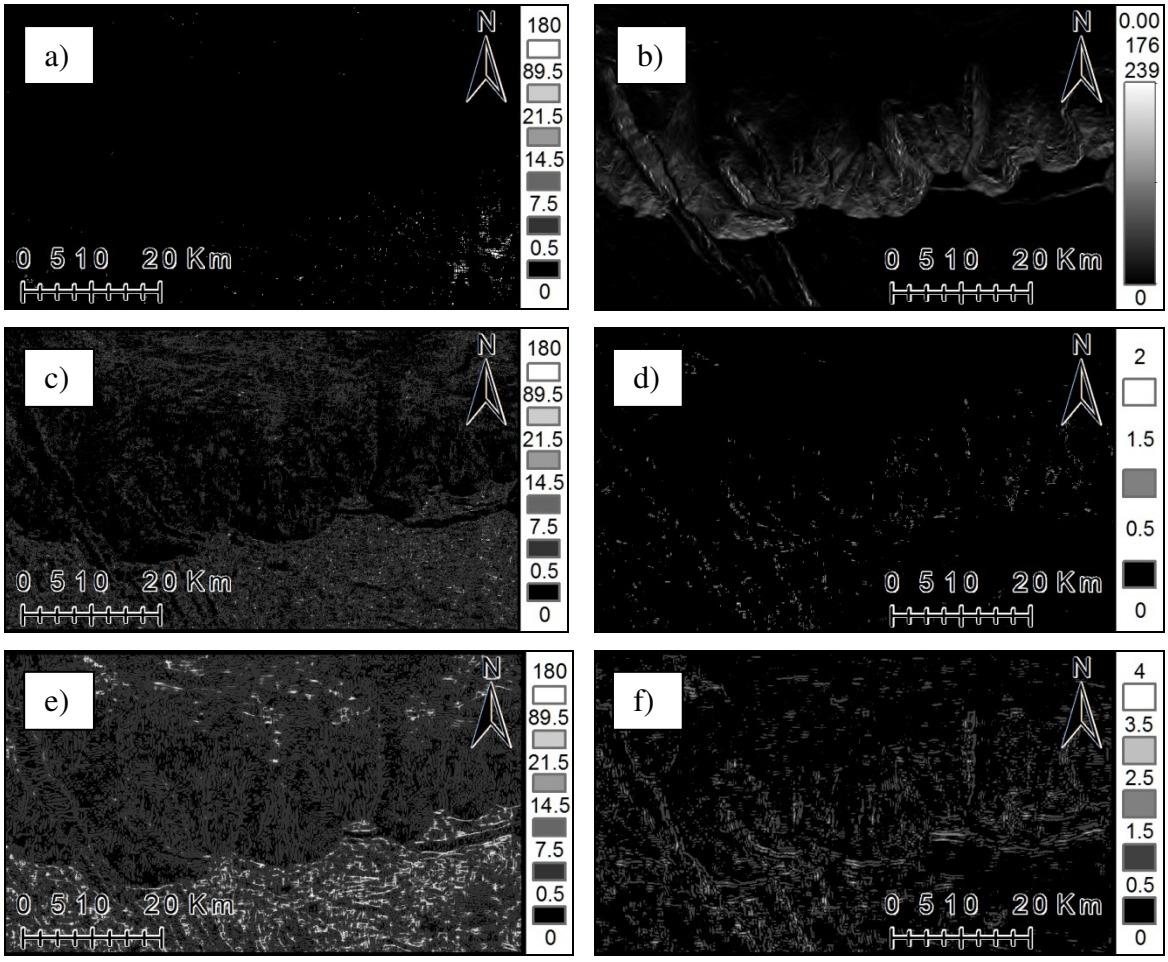


Figure 13: Difference, in degrees, in the Strike (left) and Dip (right) values between the Least Squares Regression and the Nearest Neighbour Differences
a) and b) using 3x3 moving window; c) and d) using 7x7 moving window; e) and f) using 15x15 moving window. Colour map for angular differences displayed on the left. All images were produced from differences rounded to the nearest integer, except image b), which had no difference in excess of 0.00176239. All strike colour maps were built using standard deviation breaks for the image (e) with an additional break inserted at 0.5º.

A high Mean Square Error (MSE) indicates surface anomalies such as zones of exceptional roughness, errors in the DEM, or features of high relief (scarps, troughs, ridges, peaks, pits). Unlike the built-in Neighbourhood Standard Deviation tool in Spatial Analyst extension, MSE can distinguish a smooth surface even when steeply dipping.

The Focal Statistics tool in Spatial Analyst extension computes the standard deviation of the neighbourhood mean; hence dipping surfaces, even if perfectly flat, will show high values of Neighbourhood Standard Deviation.

In addition, MSE displays surface irregularities more distinctly than the built-in ArcGIS Curvature tool (Fig. 14). Although MSE does not distinguish between concave and convex geometries, such information can be obtained from MSSD (Fig. 7).

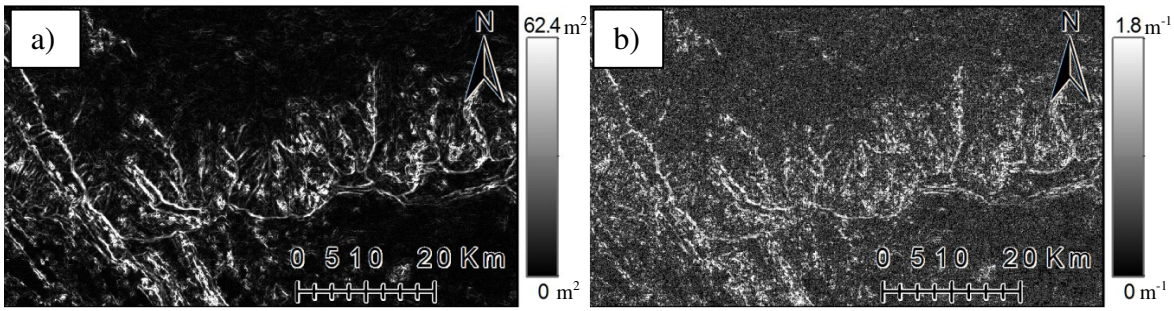


Figure 14: Sample dataset processed as a) MSE for LSR with 3x3 px moving window, and b) with built-in Curvature tool in the Surface toolset of the Spatial Analyst toolbox. Both images are displayed using the identical colour ramp, shown on the right; stretched using ArcGIS standard deviation method, n: 2.5. Since MSE is always positive, absolute values were taken of the Curvature raster to ensure consistency in representation.

The potential applications for this tool range widely, from tectonic feature and crater identification to surface roughness estimations, segmentation of raster into topographically similar domains and error elimination.

Tectonic features that have direct or erosional topographic expressions, such as grabens and faults, can be identified by their morphometric signatures present on a terrain as zones of preferred orientation. This can be achieved either by numeric analysis of gradients or of strike and dip, or by visually examining AVA for patterns. Since Cartesian plane equation parameters include the vertical offset along the common origin, they

provide an avenue to distinguish parallel surfaces from coplanar. Histograms of strike and dip can be used to determine preferred orientations in the region, as will be discussed in the following chapters.

The goodness of fit estimates can serve as indicators of surface roughness, or they can be applied for boundary identification tasks, such as detection of cliff edges and mountain ridges; e.g. on MSE, crater rims appear as nearly perfect circles (Fig. **Error! Bookmark not defined.**.a).

Complex imagery can be obtained by combining different outputs, e.g. via thresholding using Conditional tools from the Spatial Analyst toolbox (Fig. **Error! Bookmark not defined.**.b).

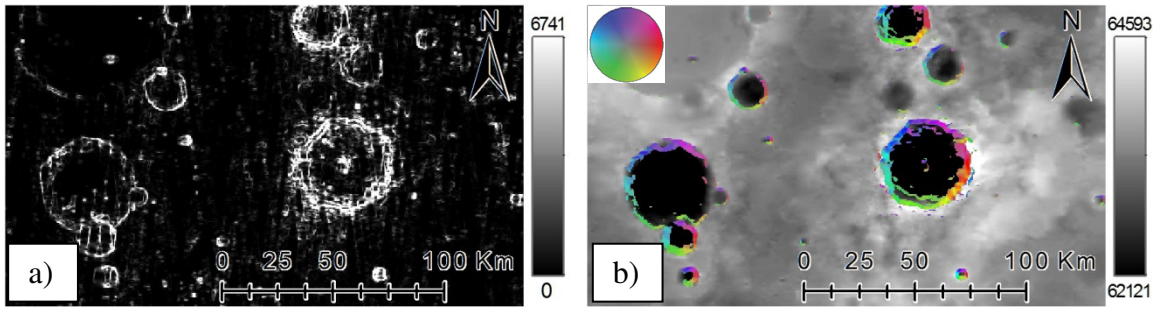


Figure 15: a) MSE obtained using a 3x3 moving window over a heavily cratered area; b) the same area displaying the DTM overlaid by AVA wherever the dip values obtained using a 3x3 window exceed 10º. AVA produced with default colour map shown in the top left corner. Source is 463m/px MOLA dataset 348°E 8°N to 352°E 12°N [http:/gis.ess.washington.edu/uwmola/]. Dataset uses false origin. Vertical unit is meter.

Since MSE can identify inappropriate values, noise can be reduced by setting to null those pixels on a DEM where MSE falls above a certain threshold (Fig. 16). Then, if desired, the resultant voids can be filled using ArcGIS built-in interpolation tools.

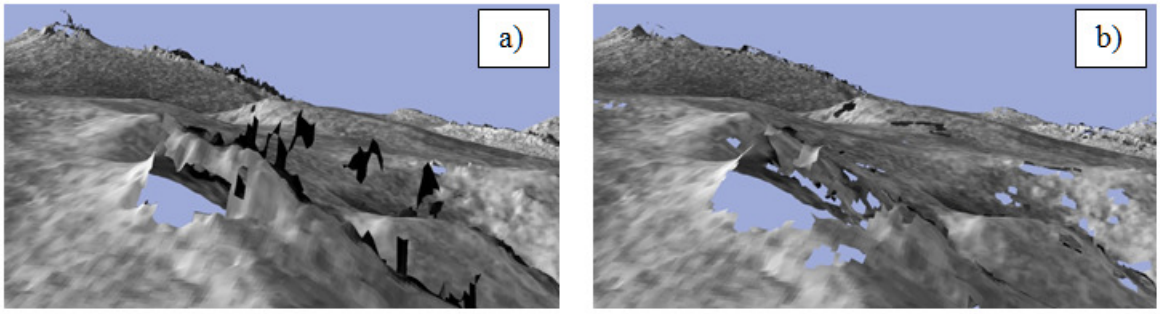


Figure 16: 3D projections of DTM for CTX stereopair ESP_033103_1785_RED-ESP_034448_1790_RED demonstrate the outcome of noise reduction. a) Spikes in the DTM are clearly the result of noise rather than true geological features; b) noise is eliminated by setting to null all pixels where LSR with a 5x5 moving window returns MSE values greater than 0.008 m.

CONCLUSION

Where geologic features such as faults or bedding planes control topography, they can be recognized using this tool, since they will appear as areas of consistent attitudes. Using the built-in ArcGIS tools, output rasters can be easily converted to a text file of Comma Separated Values, allowing for quantitative assessment and statistical analysis to be performed. This reduces the demands on the operator when searching for topographic expressions of geological structures in large datasets, while adjustable moving window size helps to compensate for surface roughness without sacrificing accuracy.

This new add-on for ArcGIS computes surface derivatives with more flexibility than the built-in tools. It allows user to define the size of the sampling region and to choose various algorithms. It can calculate the Cartesian plane equation parameters, strike, dip and four kinds of goodness-of-fit estimates. Any of these outputs can be exported in ASCII format for quantitative analysis. Additionally, this add-on can display the full 3D orientation of a surface unambiguously, by representing strike as hue and dip as saturation in the Augmented Visualization of Attitude (AVA).

Chapter 3: Topography quantification based on convolution kernels

INTRODUCTION

A small focus group tested the tool described in the previous chapter. The beta-testers suggested several improvements, such as: reducing processing time, introducing a batch processing capability and allowing selection of custom colour maps for AVA representations. Follow-up research into these recommendations led to the development of more efficient methodology implemented in the convolution tool.

Linear regression was found to be an inefficient means of calculating the gradient. Whereas Gaussian elimination itself required over 50 floating-point arithmetic operations, the major hindrance to this process is actually caused by calculating the inputs for the regression analysis, i.e. the sums $\sum_{i=1}^{N} x_i^2$, $\sum_{i=1}^{N} x_i y_i$, $\sum_{i=1}^{N} x_i$, $\sum_{i=1}^{N} y_i^2$, $\sum_{i=1}^{N} y_i$, $\sum_{i=1}^{N} z_i$, $\sum_{i=1}^{N} z_i x_i$, and $\sum_{i=1}^{N} z_i y_i$. Since the number of operations needed to calculate these sums, while insignificant for a small window size, grows proportionally to the area of a moving window, i.e. with the square of the moving window half-span (Fig. 11), it became paramount to simplify the algorithm to avoid computing these sums.

This was achieved via convolution filtering; a method that is mathematically equivalent to the least squares regression.

Proof of mathematical identity between Least Squares Regression and Convolution

The tool described in the previous chapter used the top left corner of the DEM as the coordinate system origin for all computations. However, the magnitude of the gradient, i.e. the rate of change in elevation, depends on the pixel's neighbourhood and not on a position of origin. Indeed, as pointed out in the previous chapter, the strike and dip of a plane are independent from the coefficient *d'* of its Cartesian plane equation.

Therefore, an origin can be dynamically assigned relative to the moving window, and, since $\sum_{j=-n,k=-m}^{n,m} j = 0$, $\sum_{j=-n,k=-m}^{n,m} k = 0$, and $\sum_{j=-n,k=-m}^{n,m} jk = 0$, doing so simplifies the matrix equation shown in the previous chapter to:

$$\begin{bmatrix} \sum_{j=-n,k=-m}^{n,m} (Xj)^2 & 0 & 0 \\ 0 & \sum_{j=-n,k=-m}^{n,m} (Yk)^2 & 0 \\ 0 & 0 & N \end{bmatrix} \begin{pmatrix} a' \\ b' \\ d' \end{pmatrix} = \begin{bmatrix} -\sum_{j=-n,k=-m}^{n,m} z_{j,k} Xj \\ -\sum_{i=-n,j=-m}^{n,m} z_{j,k} Yk \\ -\sum_{j=-n,k=-m}^{n,m} z_{j,k} \end{bmatrix}$$

where X and Y represent grid spacing along West-East and South-North directions; j and k correspond to the location of the pixel relative to the centre of the moving window; and m, n are the half-spans of the moving window.

The sum of squares of a natural number series can be found as [Appx. 1]

$$\sum_{t=0}^{n} t^2 = \frac{1}{6} n(n+1)(2n+1).$$

Consequently,

$$\sum_{j=-n,k=-m}^{n,m} j^2 = \frac{1}{3} n(n+1)(2n+1)(2m+1),$$

and

$$\sum_{j=-n,k=-m}^{n,m} k^2 = \frac{1}{3} m(m+1)(2m+1)(2n+1).$$

Hence, the gradients can be computed as weighted averages [Appx. 5.2] of the neighbourhood pixels:

$$a = \frac{-\sum_{j=-n,k=-m}^{n,m} z_{j,k} Xj}{\sum_{j=-n,k=-m}^{n,m} (Xj)^2} = \frac{-X \sum_{j=-n,k=-m}^{n,m} z_{j,k} j}{2(2m+1)X^2 \sum_{j=-n}^{n} j^2} = \frac{-3 \sum_{j=-n,k=-m}^{n,m} z_{j,k} j}{Xn(n+1)(2n+1)(2m+1)}$$

$$b = \frac{-\sum_{j=-n,k=-m}^{n,m} z_{j,k} Yk}{\sum_{j=-n,k=-m}^{n,m} (Yk)^2} = \frac{-Y \sum_{j=-n,k=-m}^{n,m} z_{j,k} k}{2(2n+1)Y^2 \sum_{k=-m}^{m} k^2} = \frac{-3 \sum_{j=-n,k=-m}^{n,m} z_{j,k} k}{Ym(m+1)(2m+1)(2n+1)}$$

This procedure is known as Convolution filtering, and the weight matrix is called convolution kernel [Russ, 1992]. Using ○ to represent Hadamard product, i.e., element-wise multiplication [Styan, 1973], and Σ( ) to mean summation of all elements, convolution operations can be rewritten as

$$\sum_{j=-\mathrm{n},k=-\mathrm{m}}^{n,m} z_{j,k}j = \sum(Z \circ J), \qquad \text{and} \qquad \sum_{j=-\mathrm{n},k=-\mathrm{m}}^{n,m} z_{j,k}k = \sum(Z \circ K).$$

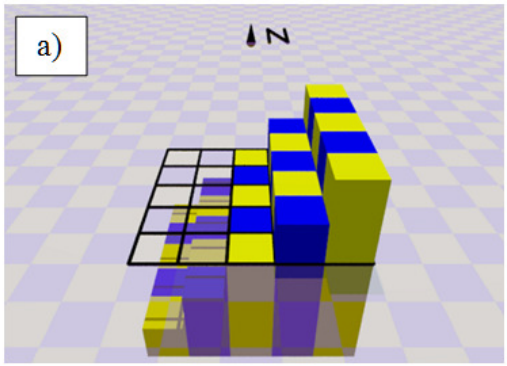
By pre-computing normalization constants

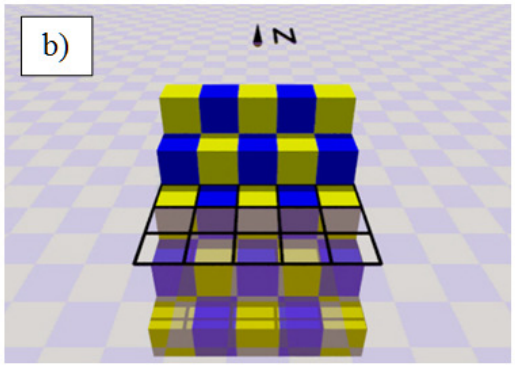
$$A = \frac{-3}{Xn(n+1)(2n+1)(2m+1)}, \qquad \text{and} \qquad B = \frac{-3}{Ym(m+1)(2m+1)(2n+1)}.$$

gradients can be calculated very rapidly:

$$a' = A\sum(Z \circ J), \qquad \text{and} \qquad b' = B\sum(Z \circ K).$$

To generate gradients in West-East and South-North directions, convolution matrices *J* and *K* need to be populated so that $J_{j,k} = j$, and $K_{j,k} = k$ (Fig. 17).





		j				
	J[j,k]	-2	-1	0	1	2
k	2	-2	-1	0	1	2
	1	-2	-1	0	1	2
	0	-2	-1	0	1	2
	-1	-2	-1	0	1	2
	-2	-2	-1	0	1	2

		j				
	K[j,k]	-2	-1	0	1	2
k	2	2	2	2	2	2
	1	1	1	1	1	1
	0	0	0	0	0	0
	-1	-1	-1	-1	-1	-1
	-2	-2	-2	-2	-2	-2

Figure 17: Convolution matrices for calculating the surface gradients, and their visualizations in a) West- East and b) South-North directions, with 5x5 px moving window, i.e. for square kernels with half-span of 2 px.

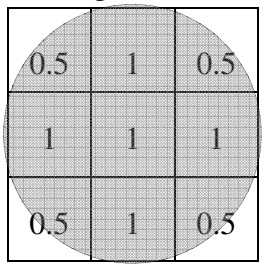
Circular kernels

For user-defined kernels, their shape can be adjusted by modifying the elements of the convolution matrix. For instance, circular kernels are commonly used in image processing to avoid distortions and are indeed required for the procedure discussed in the following chapter. Representing a circular kernel on a square moving window can be done by applying a weighting factor relative to how much of neighbourhood pixel's Voronoi region [Appx. 1] is occupied by an inscribed circle.

For example, a circle kernel inscribed in a 3x3 moving window covers side pixels almost completely, but only half of the corner pixels. The resulting shape closely resembles an octagonal kernel. Circular gradient kernels for a 3x3 moving window are generated by applying weights to the square gradient matrix elements proportional to the

area overlapped by a circle. Hence, a weighting factor of 0.5 can be applied to the corner pixels of convolution matrix (Fig. 18).

Weight matrix



West-East gradient:

-0.5	0	0.5
-1	0	1
-0.5	0	0.5

South-North gradient:

0.5	1	0.5
0	0	0
-0.5	-1	-0.5

Figure 18: The weight matrix and gradient convolution matrices for 3x3 circular kernels

Normalization constants A and B for circular kernels must be computed from weighted kernels themselves, i.e.,

$$A = \frac{1}{X \sum_{j=-n,k=-m}^{n,m} W_{j,k} \left(J_{j,k}\right)^2}, \quad \text{and} \quad B = \frac{1}{Y \sum_{j=-n,k=-m}^{n,m} W_{j,k} \left(K_{j,k}\right)^2}.$$

where $W_{j,k}$ is an element of a circle overlap weight matrix.

Computing the area overlapped by a circle for larger kernels can be done by segmenting the pixel into rectangular areas, as dictated by the extents of the curve traced on the pixel by the bounding circle, and calculating each component separately (Fig. 19).

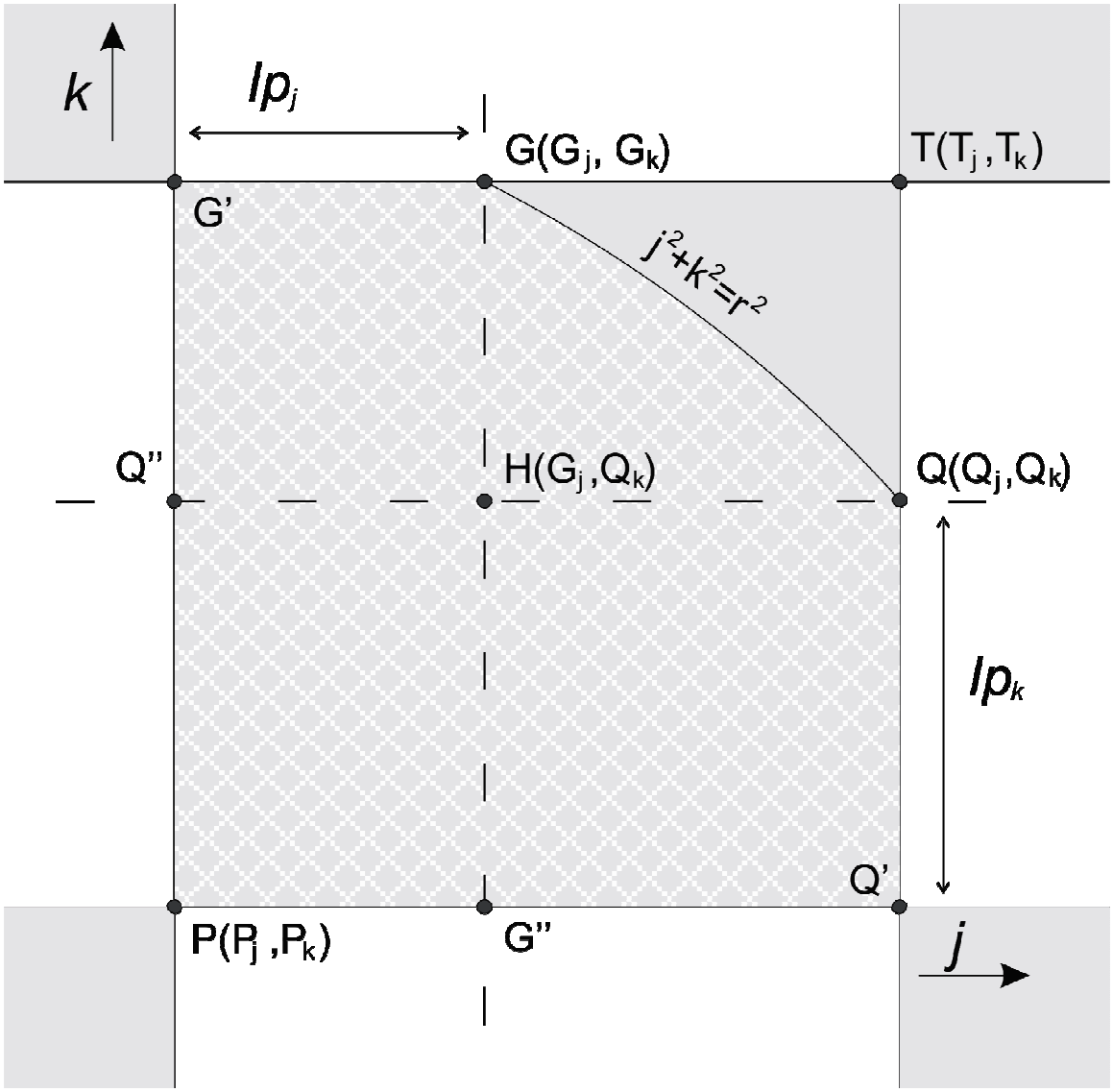


Figure 19: Calculating the area of a single pixel overlapped by a circle through segmenting it into manageable subsets.

Since circular kernel is symmetric across *J* and *K* axes, it is sufficient to calculate an array in only one quadrant, and then reflect it to obtain other values.

As illustrated in Figure 18, the area of a shape *PG'GQQ'* is:

$$A(PG'GQQ') = A(HGQ) + A(PQ''QQ') + A(PG'GG'') - A(PQ''HG'')$$

$$= A(HGQ) + Ip_j + Ip_k - Ip_j Ip_k.$$

where [http://www.integral-calculator.com/]

$$A(HGQ) = \int_{G_j}^{Q_j} \sqrt{r^2 - j^2} = \frac{1}{2}\left(r^2 \sin^{-1}\left(\frac{j}{r}\right) + j\sqrt{r^2 - j^2}\right)\Big|_{G_j}^{Q_j} - Q_k.$$

Since the pixel's size is 1x1 px by definition, for a pixel located at *(j,k)*

$$P = \left(j - \frac{1}{2}, k - \frac{1}{2}\right), \qquad \text{and} \qquad T = \left(j + \frac{1}{2}, k + \frac{1}{2}\right),$$

And, with

$$P_j = j - \frac{1}{2}, \quad P_k = k - \frac{1}{2};$$

$$P_j' = \sqrt{r^2 - \left(k + \frac{1}{2}\right)^2}, \quad P_k' = \sqrt{r^2 - \left(j + \frac{1}{2}\right)^2};$$

the values G_j, G_k and Q_j, Q_k can be computed with a help of conditional statements:

$$G_j = \begin{cases} P_j, & P_j \geq P_j' \\ P_j', & P_j < P_j' \end{cases}, \qquad G_k = \sqrt{r^2 - G_j^2};$$

$$Q_k = \begin{cases} P_k, & P_k \geq P_k' \\ P_k', & P_k < P_k' \end{cases}, \qquad Q_j = \sqrt{r^2 - Q_k^2}.$$

Also:

$$Ip_j = G_j - P_j = G_j - j + 0.5, \qquad Ip_k = Q_k - P_k = Q_k - k + 0.5;$$

Therefore, the end goal is to calculate

$$A(PG'GQQ') = \frac{1}{2}\left(r^2 \sin^{-1}\left(\frac{j}{r}\right) + j\sqrt{r^2 - (j + 0.5)^2}\right)\Big|_{G_j}^{Q_j} - (Q_j - G_j)Q_k + Ip_j + Ip_k - Ip_j Ip_k$$

Since there is always an odd number of rows and columns in a sampling window, the coordinate system origin is precisely the middle of the center pixel; therefore, the radius of an inscribed circle is r = *halfspan+0.5*; e.g., for a 11x11 px moving window, inscribed circle radius would be *r= 5.5 px*.

Care must be taken when computing the overlap area of edge pixels where *j=0* or *k=0*, due to a sign inversion across the coordinate axis. However, since these pixels are covered by the inscribed circle almost completely, their weights can be rounded to one.

It is possible to simplify the process by multiplying each kernel matrix by normalization factors A and B mentioned earlier.

Gaussian kernels

To give a greater significance to the data points closer to the centre of the sampling region, a Gaussian weight matrix may be used, as defined by the function

$$g[j,k] = c\, e^{\left(\frac{-(j^2+k^2)}{2\sigma^2}\right)},$$ (Jain et al. 1995),

where c is the normalizing constant, and σ determines the degree of smoothing (Goodchild and Fueten, 1998).

CONSIDERATIONS FOR CHOICE OF IMPLEMENTATION

Various options were considered in search of the most adequate approach to convolution filtering. Convolution subroutines do exist within ArcObjects, employed internally for neighbourhood filtering, e.g. within the Spatial Analyst toolset and as one of the ArcGIS Raster Functions. The use of existing Spatial Analyst tools, ArcGIS Raster Functions, as well as programming of a custom Geoprocessing Function were explored, but rejected in favour of a Python script evoking *NumPy* library for a number of reasons. A brief summary of these is provided next.

ArcGIS Raster Functions

Raster Functions affect the data display on the fly, and are unstable (i.e., prone to crashing) for complex processes, large datasets or when results must be exported. AVA generation using Raster Functions is unreliable even for very small datasets.

Spatial Analyst tools in Model Builder

ArcGIS provides convolution operations as part of the *Spatial Analyst* extension. It is available from within the *Neighbourhood* toolset, as the *Focal Statistics* tool with the *neighborhood* parameter set to *NbrWeight(kernel_file)*; however, this setting is not accessible from the *Model Builder*, due to a bug in the *ArcGIS 10.2.2* [Appx. 1].

Invoking Spatial Analyst tools from Python script

The aforementioned bug was found to be avoidable by calling the *Focal Statistics* tool from a python script. A python wrapper for this procedure allowed it to be included in Model Builder workflows. However, since the FocalStatistics tool is not a neighbourhood iterator, this approach limits the Sum of Squared Errors (SSE) to be only computable as a second derivative [Appx. 5.4]:

$$SSE = \sum (Measured - Predicted)^2 = \sum (z^2) - \frac{\left(\sum z\right)^2}{N} - a^2 \sum (i^2) - b^2 \sum (j^2)$$

Unfortunately, Spatial Analyst algebra is conducted at a bit depth of 32-bit. This is insufficient for the subtractions of large sums; consequently, Goodness-of-Fit estimates become compromised by floating point arithmetic rounding (Fig. 20). Correctly calculating Sum of Squared Errors via convolution demands greater bit depths, e.g. 64-bit, especially for large kernels.

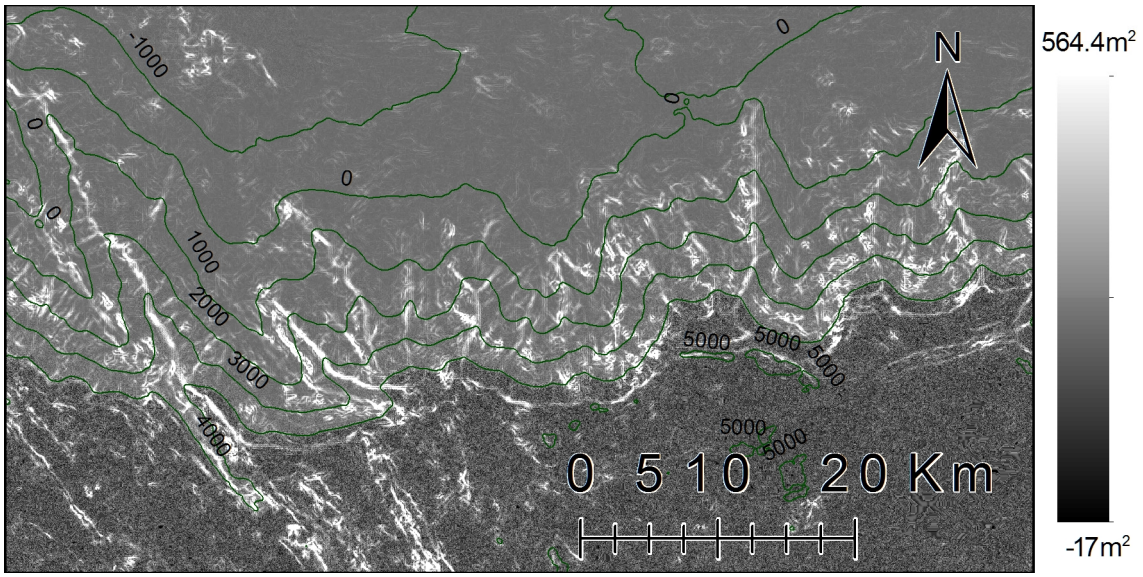


Figure 20: Sum of Squared Errors obtained with condensed sums with a 3x3 kernel; noise introduced by floating point arithmetic rounding is clearly visible. Elevation contours drawn at 1000 m intervals are overlaid in green for clarity.

Custom Geoprocessing Function

Accessing convolution subroutines through a Custom Geoprocessing Function has been successfully attempted. It was written as an extension implementing *IGPFunction2* and IGPFunctionFactory interfaces [Appx. 1, Appx. 2]. Using the *IGP* (geoprocessor) interface family made this tool available from Model Builder, similarly to other geoprocessing tools in ArcToolbox. That, in turn, made it possible to create complex workflows and conduct batch operations. However, installing this function required the use of the *ESRIRegAsm* utility, which directly modifies the Windows Registry database. Correct application of the *ESRIRegAsm* utility demands advanced knowledge of Operating System administration and thus is unsuitable for average users.

While ESRI has developed Visual Studio project templates containing framework code necessary for creating installers responsible for automatic registration of

geoprocessing functions, these templates are not part of ArcGIS online distribution, and, as such, were not available for this work. Without the necessary templates, programming an installer became unfeasible.

Invoking NumPy library from Python script

ArcGIS ships with a powerful *NumPy* (numeric python) library that provides handles for robust Fortran-based array computing functions. Data can be converted between raster and *numpy.array* formats via *NumPyArrayToRaster* and *RasterToNumPyArray* functions from the built-in *arcpy* library, thus altogether eliminating the need for the *Spatial Analyst* extension. Python scripts do not require installation and they can be imported into ArcGIS without administrator privileges.

The *Spatial Analyst* extension requires an additional licence beyond ArcMap, while NumPy is an open source library. Hence, the scripts written for NumPy can even be applied outside of ArcMap, for as long as the raster data can be converted to a NumPy array. Such conversion can be achieved with a variety of freely available libraries, e.g. *libtiff*. However, raster processing outside of ArcMap environment was not explored, since this subject falls beyond the scope of the thesis, and *Spatial Analyst* extension was still used in most of the tools.

Upon evaluating the options considered, the NumPy approach was deemed the optimum choice for implementation of convolution filtering.

IMPLEMENTATION OF CONVOLUTION FILTERING

The working implementation is a python script, simply installable by right-clicking *ArcToolbox* window and choosing "*Add Toolbox…*", then selecting this toolbox on the disk. The Toolbox contains *Convolution* and *AVA* scripts, as well as ArcGIS Models for calculating Strike and Dip from gradient. It also contains an *All-In-One* script to produce in one run some or all of these rasters: *South-North* and *West-East* gradients, Strike, Dip, Goodness-of-Fit and AVA.

Convolution script

Since no usable tool exists in ArcGIS to perform convolution filtering with a user-defined kernel, a new python script was written specifically for this task (Fig. 21). This tool does not require Spatial Analyst extension to run.

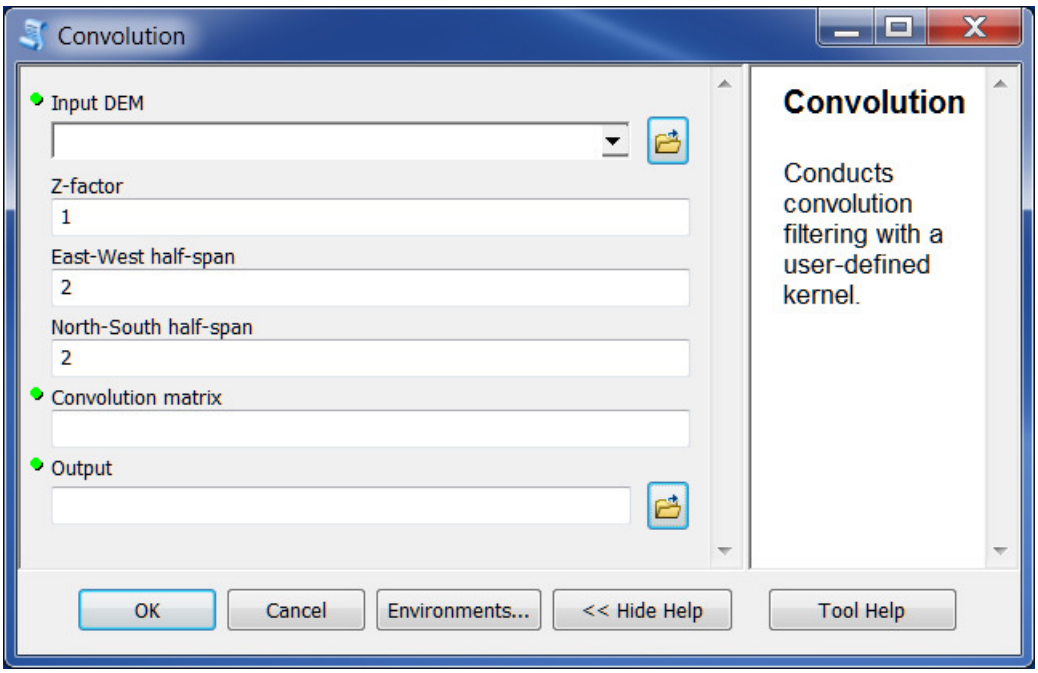


Figure 21: Convolution tool dialog. Z factor is the vertical exaggeration of the DTM.

Custom matrices must be supplied to this tool as a string of comma-delimited text characters. This string can be generated in Excel (Fig. 22), then copied and pasted into the user interface. Examples of convolution kernels for gradients are shown in Figure 23.

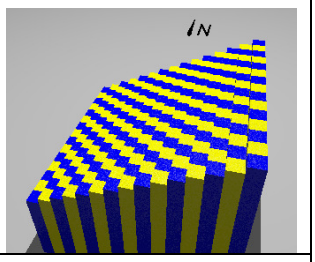
	A	B	C	D	E	F
1		-2	-1	0	1	2
2	-2	=B$1	-1	0	1	2
3	-1	-2	-1	0	1	2
4	0	-2	-1	0	1	2
5	1	-2	-1	0	1	2
6	2	-2	-1	0	1	2

=B2-AVERAGE($B$2:$F$6)	-1	0	1	2
-2	-1	0	1	2
-2	-1	0	1	2
-2	-1	0	1	2
-2	-1	0	1	2

Figure 22: Computing 5x5 convolution matrix for E-W gradient in Excel

To maximize flexibility, the convolution script does not take into account grid spacing. It simply returns a weighted sum of the neighbourhood. To ensure appropriate results when computing strike and dip, output gradients must either be multiplied by a correct normalization coefficient, or be produced with the kernels that have normalization coefficient already factored in. It is left up to the user to ensure that the script is supplied with a correct string of characters, since no input validation takes place.

a) E-W	**-2**	**-1**	**0**	**1**	**2**
-2	-2	-1	0	1	2
-1	-2	-1	0	1	2
0	-2	-1	0	1	2
1	-2	-1	0	1	2
2	-2	-1	0	1	2



b) N-S	**-2**	**-1**	**0**	**1**	**2**
-2	2	2	2	2	2
-1	1	1	1	1	1
0	0	0	0	0	0
1	-1	-1	-1	-1	-1
2	-2	-2	-2	-2	-2

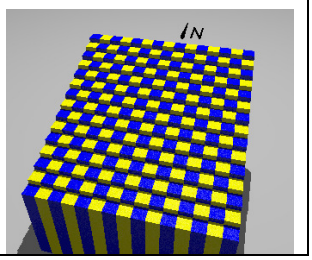


Figure 23: Convolution matrices shown for 5x5 kernels for (a) East-West gradient, (b) North-South gradient and their corresponding geometry, using larger kernels for better visualization.

AVA script

The AVA script tool creates AVA visualizations using inputs of integer rasters, such as the – Strike Raster with a range of 0 to 359, and the Dip raster with a range of 0 to 90. Improved upon the previous version, it, in addition to the standard approach, also allows using custom colour maps. (Fig. 24). Colour map is simply a 360x90 pixel RGB image where x-direction represents the strike, and y-direction represents the dip.

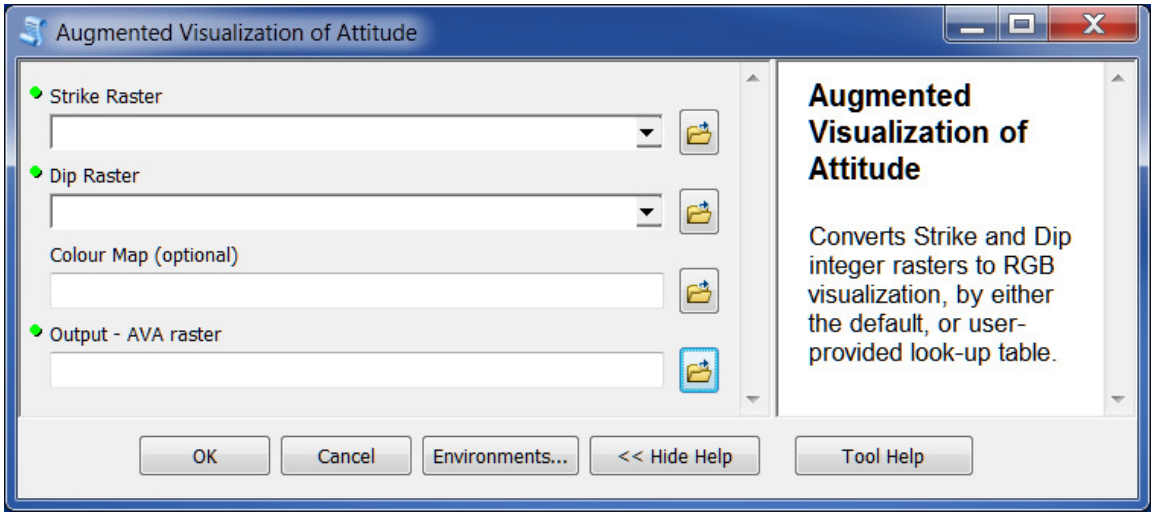


Figure 24: AVA script tool

All-In-One script

This script takes the following inputs: the DEM, the vertical exaggeration (i.e., the z-factor) and the kernel South-North and West-East half-spans. All of these parameters are required for computation, and all but the DEM have a default value of "1". The output parameters are all optional; however, at least one must be entered to run the script. The last parameter allows for selecting other than the default colour map for AVA (Fig. 25).

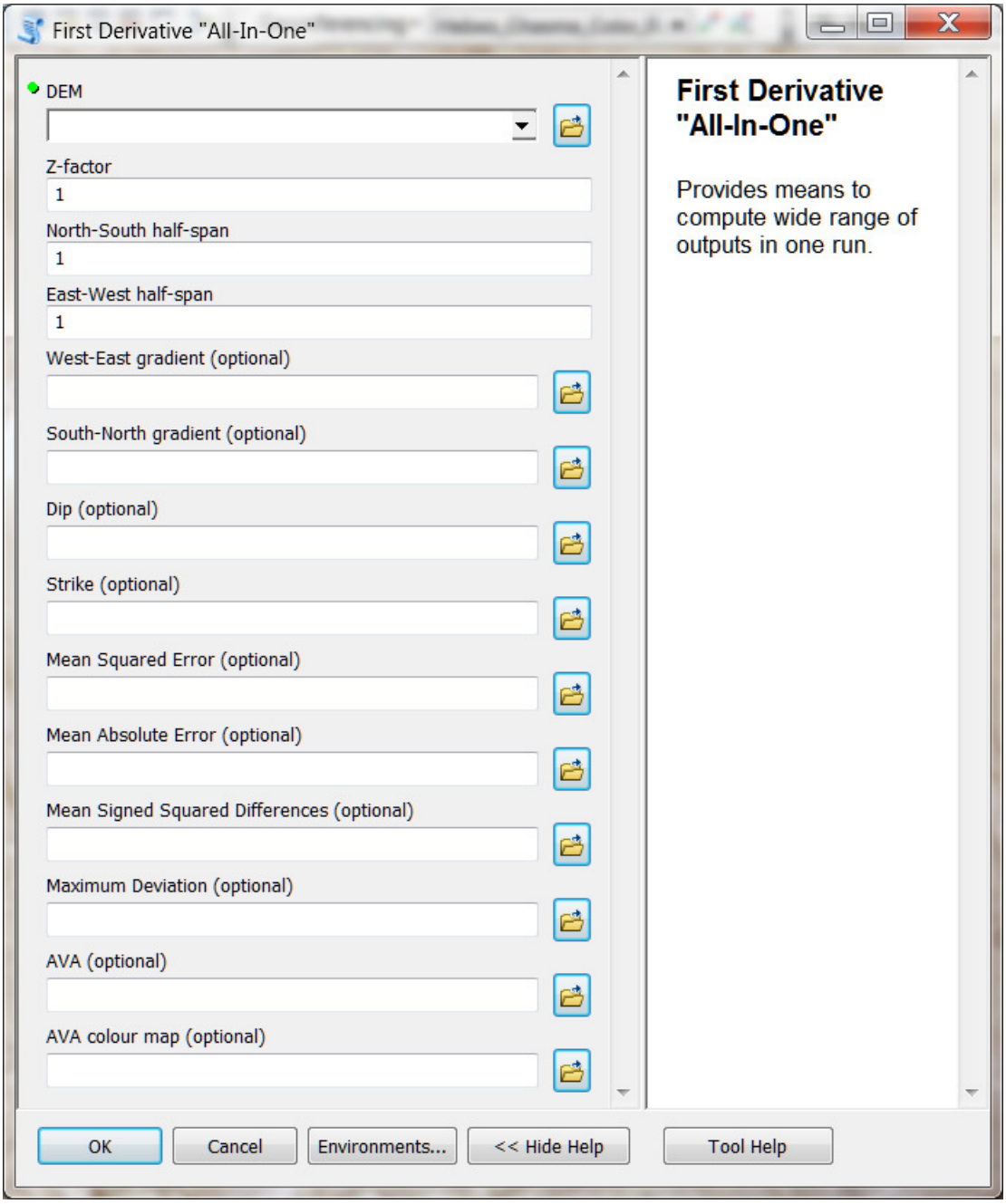


Figure 25: First Derivative All-In-One python script dialog box

In this work, the All-in-One script was only designed to apply square kernels, as they are easier to implement than the circular ones; however, kernels of other shapes can be applied using Convolution script tool.

CODE OVERVIEW

All scripted tools presented in this chapter are architecturally similar.

Imported modules and input collection

The *All-In-One* script imports *numpy*, *arcpy* and *os* modules. The module *numpy* is needed for mathematics and array manipulations. The module *os* is used for separating file path into file name and folder address. The module *arcpy* is used for collecting user inputs, converting between raster and numpy arrays, as well as saving the output rasters. A function *arcpy.GetParameterAsText(index as int)* provides access to user inputs.

Defining pixel blocks

There exists a memory limitation arising from using a 32-bit address space, which limits maximum accessible memory to 2 GB. [Appx.1] This limit can be overcome by installing a 64-bit background geoprocessing extension. Since it was decided to avoid relying upon additional installations, the memory limitation was solved in a manner similar to the approach taken in the previous chapter, i.e., by splitting the input raster into blocks, processing each block separately, and then re-assembling them upon completion. Consequently, the DEM is subdivided into 2^{11} by 2^{11} px blocks. The irregular sized blocks positioned on raster edges are obtained as remainders of this subdivision. Thus, four sets of tiles of equal dimensions are produced. Each set is then processed separately, and all are recombined using *Raster Catalog To Raster Dataset* built-in function from *Data Management* toolbox.

Testing the tool on large datasets revealed that creation of a numpy array within ArcGIS results in a memory leak. The operative memory (Random Access Memory), allocated for every new numpy array, is not released when that array goes out of scope. This is due to a system bug causing incorrect incrementing of the object reference variable. Thus, creating new arrays for every pixel block quickly runs down the system's resources and induces ArcGIS to crash. A workaround to this problem was found in reusing arrays. Such recycling is possible since each tile within a set has the same size. The script iterates over all tiles in a set, saving them to the workspace. The current tile number is displayed in messages, so that the progress can be monitored.

Workflow for each tile

Processing each tile begins with multiplying the input data by a z-factor scaling, i.e. reciprocal of vertical exaggeration, so that vertical and horizontal coordinate axes have the same units. This facility is important, since their units are not always the same, e.g. a high accuracy DEM may have horizontal units in meters, and vertical units in decimeters.

For the next step, a convolution operation is run to find the South-North and West-East gradients. In the All-In-One script the weights for the convolution matrix, to increase computational efficiency, are supplied in map units rather than pixel displacement. The Convolution script uses the weights exactly as provided by the user. The Convolution script then proceeds to save the pixel block, while the All-In-One script may additionally call upon functions for calculating other representations: strike, dip, goodness of fit, and AVA, provided such outputs have been requested by the user.

Computing the goodness of fit is very similar for each of the available options and is achieved by first calculating the mean value for the kernel, then using it together with gradient to determine the deviation of its every point from the predicted value.

The conversion to AVA is achieved through the *numpy.take()* function, which matches pixel values to indices of a reference colour map.

RESULTS

Strike and dip measurements of a 1516x875 px error-free sample dataset, as obtained via convolution with All-In-One script, were subtracted from the measurements obtained with 15x15 px square sampling window Least Squares Regression rounded to an integer. No differences were found, except when LSR value ended exactly at .5. Out of the 129329 pixels on the sample raster dataset only 8 were 1 degree less in LSR than Convolution. This discrepancy was caused by differences in the rounding algorithm, since the All-In-One script rounds to a nearest integer, an option unavailable in ArcGIS, so LSR was rounded via adding of 0.5 followed by trimming the decimals.

Convolution with NumPy yields results identical to the LSR tool; however, it is much faster (Fig. 26).

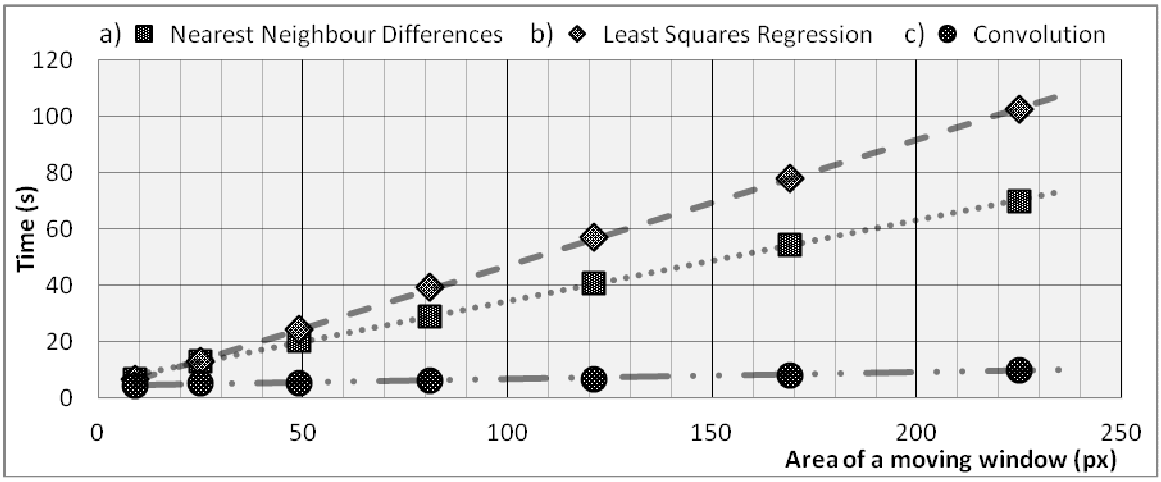


Figure 26: Computing times and their lines of best fit for: a) NND, b) LSR, and c) the All-In-One convolution script, when run for making AVA only, plotted against the area of a moving window, when run with a square kernel on a 1516x875px DEM.

Processing raster with circular rather than square kernel produced only up to 1° differences, nearly randomly distributed, a value within the ±1° margin of error (Fig 27a). Square kernel appears to give a slightly higher numbers consistently for larger kernels (Fig 27b).

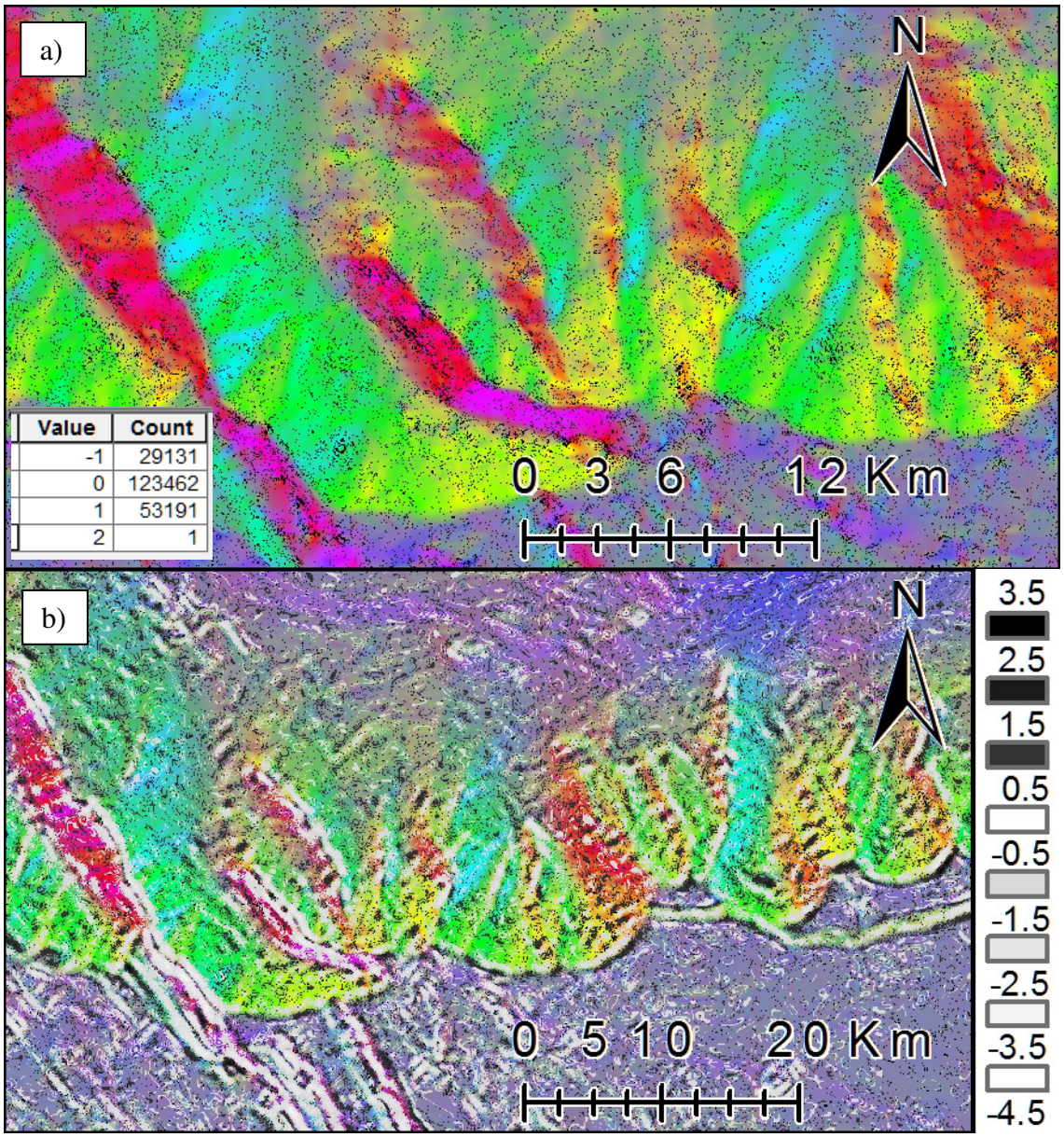


Figure 27: Dip rasters discrepancy, in degrees, from circular kernel subtracted from square kernel for a) 5x5px moving window, with attribute table for differences raster inserted in the bottom left corner, and b) 23x23px window, with legend on the right.

Since there is barely any difference in output, using a more complex technique of circular kernels would generally be unnecessary when working with first derivative.

DISCUSSION

Convolution methods are identical to the non-regression methods of computing surface derivatives described in introduction to Chapter 2:

1. For a 2x2 kernel, the convolution method is identical to the second-order finite difference method on a 2x2 moving window proposed by Dozier and Strahler [1983], for every pixel not on the image edge.
2. On a 3x3 square kernel, convolution is identical to the third-order finite difference method without a weighting factor, as proposed by Sharpnack and Akin [1969].
3. On a 3x3 circular kernel, it is identical to the method proposed by Horn [1981], which is currently being used as the ArcGIS algorithm for calculating Aspect and Slope in the Spatial Analyst extension.

Sometimes it may be useful to look at the undulations, while disregarding regional trends. Preference for high frequency data can be obtained by subjecting the DEM to a High-Pass filter prior to calculating the derivative (Fig. 28.b). Alternatively, signals within a specific frequency range can be extracted by calculating the disparity between gradients obtained with different kernel sizes (Fig. 28.c). The AVA obtained from the difference of 3x3 and 5x5 gradients shows noticeably higher sensitivity towards small features than the AVA of a 3x3 gradient taken from the same DEM preprocessed using a High-Pass filter.

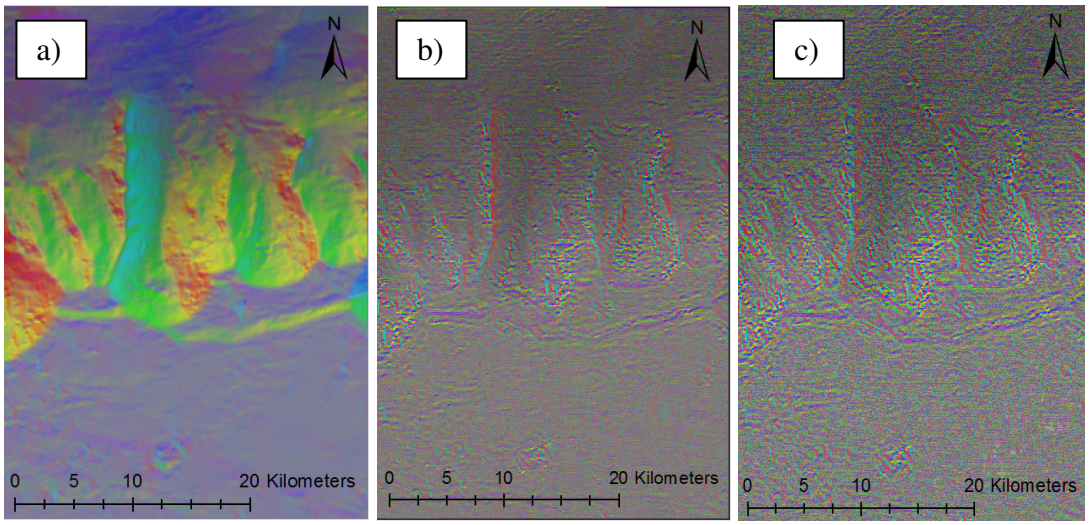


Figure 28: a) AVA obtained from a 3x3 gradient; b) AVA from a 3x3 gradient over a High-Pass filtered DEM; c) AVA of high-frequency signal, obtained by subtracting a 5x5 gradient from a 3x3 gradient.
All three images are displayed with 40% transparency over a greyscale DEM. DEM is displayed with the ArcGIS hillshading effect.

It should be noted that Python scripts must be run via ArcModel to have outputs added to the Table of Content automatically.

In addition, the custom Convolution tool, unlike LSR, cannot fill holes in the DEM. If data is incomplete, one null pixel invalidates the entire kernel for convolution. Another downfall of the Convolution method is that it does not provide a means to distinguish parallel surfaces from coplanar ones.

CONCLUSION

We have developed an ArcGIS script tool that uses a convolution to calculate the attitudes of planar surfaces using the NumPy library. Convolution with NumPy python scripts yields results similar to the Least Squares Regression available with the add-on presented in Chapter 2. These scripts are faster, customizable, and can be used for batch processing. In addition we have developed a method for using circular kernels which

allows to avoid artefacts caused by uneven kernel shape. The toolbox does not require installation, and it can be readily added to an .mxd document by any user. The scripts can be readily modified.

Chapter 4: Curvature

INTRODUCTION

The first derivative of a function is the slope. The second derivative is the rate of change of the slope, i.e. curvature. Quantifying the curvature is important, as many geological features result in curved topography. Similarly, by looking at the rate of change of the curvature, higher order derivatives can be produced.

Curvature in ArcGIS

The ArcGIS Spatial Analyst contains a tool called *"Curvature"*. It is primarily intended for the study of runoff on a drainage basin, e.g. for modelling of erosional or hydrologic processes. By employing the methodology similar to that described by Zevenbergen and Thorne [1987], this tool can generate up to three derivative rasters:

1. *Profile Curvature* is the rate of change of the slope in the direction of maximum slope, and thus distinguishes between acceleration and deceleration of flow.
2. *Planform Curvature* is measured orthogonal to the Profile Curvature and thus distinguishes between convergence and divergence of flow.
3. *Average Curvature* is the average value of *Profile* and *Planform* curvatures, and distinguishes between a peak and a sink.

Upwardly convex geometry is indicated by positive values for *Average* and for *Planiform Curvature*, and by negative values for *Profile Curvature* (Fig. 29). The size of the sampling window cannot be changed from the default 3x3 px.

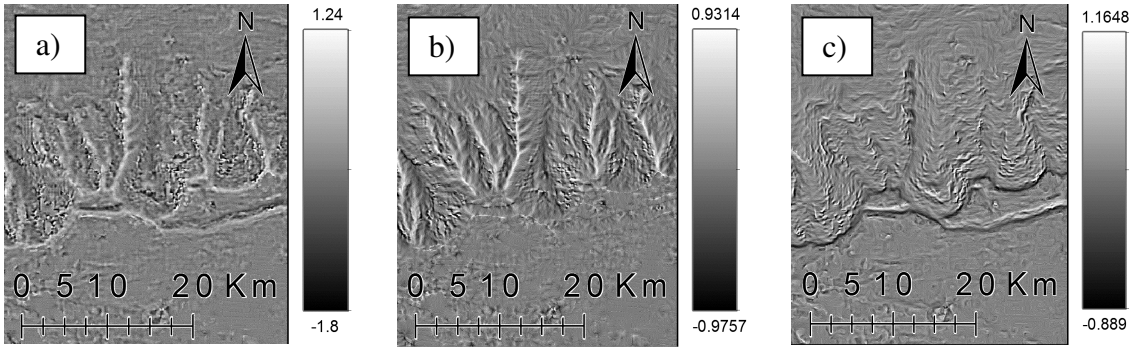


Figure 29: Outputs of *Spatial Analyst Curvature* tool, displayed with 0.5% clip: (a) *Average Curvature*, (b) *Planform Curvature*, (c) *Profile Curvature*. Units are 0.01 m^{-1}.

Curvature in Mathematics

Since the second derivative is a gradient of a gradient, the most obvious method of calculating it is recursion. A direct method for calculating curvature will be discussed later in the chapter.

Since there are two principal directions in which change can be measured, it is possible, by recursive application of convolution for first derivative to calculate second derivative terms d^2z/dxy, d^2z/dyx, d^2z/dx^2, and d^2z/dy^2. However, by Schwarz's theorem on symmetry of second derivatives [Galbis and Maestre, 2010], the mixed partials $\frac{d^2z}{dxy}$ and $\frac{d^2z}{dyx}$ are equivalent. Consequently, a 2D scalar field, such as a DEM, only has three derivatives of the second order. These are the partial derivatives in X and in Y, and one mixed derivative. The second order partial derivatives are standard basis curvatures, whereas the mixed partial is a geodesic torsion [James and James, 1992] in the standard basis direction [Anton and Rorres, 2010].

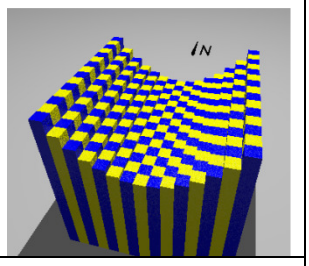
Geodesic torsion indicates the sense of rotation around the motion vector when travelling in one of standard basis directions. The positive values of geodesic torsion mean that the tangent surface rotates counterclockwise in the direction of movement, i.e.

if a cart was pushed along one of the geodesic directions, it would tilt to the left. Interestingly, the sense of rotation is identical for opposite directions, e.g. it is the same whether viewed from north to south, or vice versa.

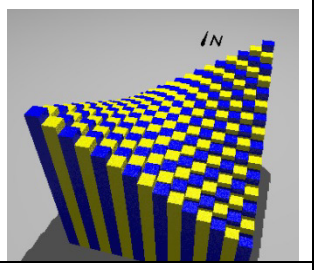
Convolution for curvatures

Similar to computing the slope, curvatures too can be found via convolution. Since convolution is a pattern-matching technique, correct matrices are with the kernels $g_A(x,y) = x^2, g_B(x,y) = xy, g_C(x,y) = y^2$ (Fig. 30)

a. X^2	-2	-1	0	1	2
-2	4	1	0	1	4
-1	4	1	0	1	4
0	4	1	0	1	4
1	4	1	0	1	4
-2	4	1	0	1	4



b. X*Y	-2	-1	0	1	2
-2	4	2	0	-2	-4
-1	2	1	0	-1	-2
0	0	0	0	0	0
1	-2	-1	0	1	2
2	-4	-2	0	2	4



c. Y^2	-2	-1	0	1	2
2	4	4	4	4	4
-1	1	1	1	1	1
-2	0	0	0	0	0
-1	1	1	1	1	1
2	4	4	4	4	4

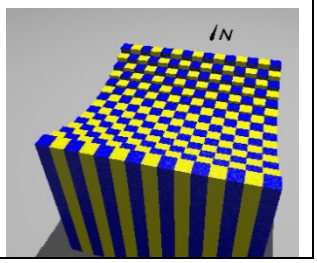


Figure 30: Convolution matrices for 5x5 kernels: (a) {XX} from X^2, (b) {XY} from XY, and (c) {YY} from Y^2, Visualizations of kernel geometry are shown to the right of each matrix.

Examination of the matrices thus produced makes it immediately apparent that {XX} and {YY} do not add up to zero. Furthermore, {XY} is a covariance term. Since covariance is greater with increasing overlap, attempting to compute the gradient using square kernels would overemphasize signals from features oriented at ±45º. Consequently, unlike the first derivative, the second derivative demands the use of circular kernels to avoid sampling bias.

Normalizing torsion to curvature

Convolution outputs need to be normalized to insure conformance between geodesic directional curvatures and geodesic torsion. Values obtained with different kernel sizes and different grid spacing should be comparable with each other as well.

Weighting coefficients were found using least squares regression, by solving a matrix equation.

$$\begin{bmatrix} \sum x^2 & \sum xy & \sum x^3 & \sum x^2y & \sum xy^2 & \sum x \\ \sum xy & \sum y^2 & \sum x^2y & \sum xy^2 & \sum y^3 & \sum y \\ \sum x^3 & \sum x^2y & \sum x^4 & \sum x^3y & \sum x^2y^2 & \sum x^2 \\ \sum x^2y & \sum xy^2 & \sum x^3y & \sum x^2y^2 & \sum xy^3 & \sum xy \\ \sum xy^2 & \sum y^3 & \sum x^2y^2 & \sum xy^3 & \sum y^4 & \sum y^2 \\ \sum x & \sum y & \sum x^2 & \sum xy & \sum y^2 & \sum N \end{bmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \end{pmatrix} = \begin{bmatrix} -\sum zx \\ -\sum zy \\ -\sum zx^2 \\ -\sum zxy \\ -\sum zy^2 \\ -\sum z \end{bmatrix}$$

With the origin taken at the centre of moving window, most elements cancel out, and geodesic torsion then appears as:

$$d = \frac{-\sum zxy}{\sum x^2y^2} = \frac{\{XY\}}{\sum x^2y^2};$$

$$\begin{bmatrix} \sum x^4 & \sum x^2y^2 & \sum x^2 \\ \sum x^2y^2 & \sum y^4 & \sum y^2 \\ \sum x^2 & \sum y^2 & N \end{bmatrix} \begin{pmatrix} c \\ e \\ f \end{pmatrix} = \begin{bmatrix} -\sum zx^2 \\ -\sum zy^2 \\ -\sum z \end{bmatrix}$$

Provided that the kernel is symmetric, $\sum x^4 = \sum y^4$ and $\sum x^2 = \sum y^2$.

The sums of elements within a square kernel are easily computable [Appx. 1]:

$$\sum y^2 = \sum x^2 = \frac{1}{3}n(n+1)(2n+1)^2, \quad N = (2n+1)^2,$$
$$\sum x^2y^2 = \frac{1}{9}n^2(n+1)^2(2n+1)^2;$$

thus, $\sum x^2y^2 = {(\sum x^2)^2}/{N} = {(\sum y^2)^2}/{N}$. Multiplying the bottom row by ${\sum x^2}/{N}$

$$\begin{bmatrix} \sum x^4 & \sum x^2y^2 & \sum x^2 \\ \sum x^2y^2 & \sum y^4 & \sum x^2 \\ \sum x^2y^2 & \sum x^2y^2 & \sum x^2 \end{bmatrix} \begin{pmatrix} c \\ e \\ f \end{pmatrix} = \begin{bmatrix} -\sum zx^2 \\ -\sum zy^2 \\ -\frac{\sum x^2}{N}\sum z \end{bmatrix}$$

allows for elimination,

$$c\left(\sum x^4 - \sum x^2y^2\right) = \frac{\sum x^2}{N}\sum z - \sum zx^2 = -\sum z\left(x^2 - \overline{x^2}\right)$$
$$e\left(\sum y^4 - \sum x^2y^2\right) = \frac{\sum y^2}{N}\sum z - \sum zy^2 = -\sum z\left(y^2 - \overline{y^2}\right)$$

From which solution can be formulated:

$$c = -\frac{\sum z\left(x^2 - \overline{x^2}\right)}{\sum x^4 - \sum x^2y^2}, \quad d = \frac{-\sum zxy}{\sum x^2y^2}, \quad e = -\frac{\sum z\left(y^2 - \overline{y^2}\right)}{\sum x^4 - \sum x^2y^2}$$

Or

$$c = -\sum(G_{xx} \circ Z), \quad d = -\sum\left(G_{xy} \circ Z\right), \quad e = -\sum\left(G_{yy} \circ Z\right)$$

Where G_{xx}, G_{xy}, G_{yy}, are convolution matrices

$$G_{xx} = \frac{g_A - \overline{x^2}}{\sum(x^4) - \sum(x^2y^2)} \quad G_{yy} = \frac{g_C - \overline{y^2}}{\sum(x^4) - \sum(x^2y^2)} \quad G_{xy} = \frac{g_B}{\sum(x^2y^2)}$$

The convolution kernels for XX and YY, unlike that for covariance term XY, do not add up to zero. Hence, these should be normalized by subtracting the average value of elements in the matrix from each element in convolution matrix, as well as multiplying the output by normalization constant.

Interestingly, the denominator is two orders less than the nominator, so normalizing for grid spacing would require dividing curvatures by spacing squared. The kernel therefore must be circular in terms of pixels, not necessarily in map units.

Taylor series expansion

In concord with Taylor-Maclaurin series expansion

$$f(x,y) = \frac{f(0,0)}{0!} + \frac{\sum f'(x,y)}{1!} + \frac{\sum f''(x,y)}{2!} + \cdots$$

A surface at a point can be approximated as a second order polynomial. The DEM, for as long as it is continuously differentiable, will have the same second partial derivatives as a quadratic equation of paraboloid surface $Z=Ax^2+Bxy+Cy^2$, where $A = \frac{d^2z}{dx^2}$; $B = \frac{d^2z}{dxy}$; $C = \frac{d^2z}{dy^2}$. Shapes whose derivatives are equal at a point are called osculating. It is thus possible to describe the second derivative of a DEM at each pixel in terms of parameters that describe its osculating paraboloid.

Eigenvalues are principal curvatures

For a quadratic in standard position the mixed derivative is zero (i.e., B=0), whereas non-zero values indicate rotation of the axis of symmetry away from coordinate system basis. Finding principal curvatures of an osculating paraboloid is thus possible by applying a rotation to eliminate the cross derivative [Anton and Rorres, 2010].

The principal curvatures are eigenvalues of

$$Z = Ax^2 + Bxy + Cy^2 = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} A & \frac{1}{2}B \\ \frac{1}{2}B & C \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix},$$

the characteristic polynomial of which is

$$\begin{vmatrix} \lambda - A & \frac{1}{2}B \\ \frac{1}{2}B & \lambda - C \end{vmatrix} = (\lambda - A)(\lambda - C) - \left(\frac{1}{2}B\right)^2,$$

and the discriminant for the eigenvalues is $\Delta\lambda = (A + C)^2 - 4 * \left(A * C - \left(\frac{1}{2}B\right)^2\right)$.

Hence, the eigenvalues are:

$$\lambda_{max} = \frac{A + C + \sqrt[2]{A^2 + B^2 + C^2 - 2AC}}{2}; \lambda_{min} = \frac{A + C - \sqrt[2]{A^2 + B^2 + C^2 - 2AC}}{2}.$$

Invariance under rotation

Notably, $\frac{\lambda_1+\lambda_2}{2} = \frac{A+C}{2} = Mean\ Curvature$, which means that the mean curvature is invariant under the rotation of paraboloid, i.e. it stays the same regardless of the choice of reference coordinates.

Elevation contours trace conic sections

The principal curvatures are reciprocals of the signed squared radius of curvature, since, rotated to principal coordinates, the paraboloid is described by:

$$Z = \lambda_1 x^2 + \lambda_2 y^2 = \frac{x^2}{r_x * |r_x|} + \frac{y^2}{r_y * |r_y|}$$

The traces of the hyperboloid on the planes *zx* and *zy* are parabolas, and the r_x and r_y are the radii of curvature of these parabolas. The smaller is the radius of curvature, the larger the eigenvalue and the more rapidly the surface is changing slope. If both eigenvalues have the same sign, the approximation is a flattened elliptic paraboloid of rotation. If both eigenvalues are positive, the paraboloid is convex; and it is concave if they are both negative. Such a paraboloid surface would trace elliptic contours; hence, it is called elliptical paraboloid. If, on other hand, one radius is positive and another negative, the surface would be saddle-shaped, and would trace a hyperbola on the plane parallel to *xy*; hence it is called hyperbolic paraboloid.

Sign of curvature

Positive curvature corresponds to concave shape; hence, rasters of λ_{max} display preferentially troughs, while λ_{min} rasters display ridges. From principal curvatures, the Gaussian curvature can be found. The Gaussian curvature is a product of principal curvatures, and indicates whether the surface is locally elliptic or hyperbolic. Rasters for minimum and maximum eigenvalues can be rerun as inputs to the first derivative tool.

This will show preferred orientations of ridges and troughs, respectively, technically as a third derivative.

Minor and major principals

By comparing absolute values of principal curvatures, we can determine the curvature along the major or minor axis of the paraboloid.

$$\begin{cases} |\lambda_{max}| > |\lambda_{min}| \;\rightarrow\; \lambda_{major} = \lambda_{max}\,, \quad \lambda_{minor} = \lambda_{min}; \\ |\lambda_{max}| \le |\lambda_{min}| \;\rightarrow\; \lambda_{major} = \lambda_{min}\,, \quad \lambda_{minor} = \lambda_{max}. \end{cases}$$

Eigenvectors and paraboloid rotation

For consistency with the strike measurement, the degree of rotation of the principal axis of a symmetric tensor was also measured clockwise from the North. The degree of rotation is found from eigenvector $v = \begin{bmatrix} v_x \\ v_y \end{bmatrix}$, as $\theta = \tan^{-1}\frac{v_x}{v_y}$.

$$\begin{bmatrix} A & \frac{1}{2}B \\ \frac{1}{2}B & C \end{bmatrix}\begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} \lambda * v_x \\ \lambda * v_y \end{bmatrix} \quad => \quad \begin{bmatrix} A * v_x + \frac{1}{2}B * v_y \\ \frac{1}{2}B * v_x + C * v_y \end{bmatrix} = \begin{bmatrix} \lambda * v_x \\ \lambda * v_y \end{bmatrix}$$

Dividing through by v_y

$$\begin{bmatrix} A * \frac{v_x}{v_y} + \frac{1}{2}B \\ \frac{1}{2}B * \frac{v_x}{v_y} + C \end{bmatrix} = \begin{bmatrix} \lambda * \frac{v_x}{v_y} \\ \lambda \end{bmatrix} \quad => \quad \lambda = \frac{1}{2}B * \frac{v_x}{v_y} + C.$$

Hence, $\frac{v_x}{v_y} = \frac{\lambda - C}{\frac{1}{2}B}$; and $\theta = \mathrm{Tan}^{-1}\left(\lambda - C, \frac{1}{2}B\right)$.

Alternatively:

$$\begin{bmatrix} \frac{1}{2}Bv_y \\ \frac{1}{2}Bv_x \end{bmatrix} = \begin{bmatrix} \lambda v_x - Av_x \\ \lambda v_y - Cv_y \end{bmatrix} => \frac{1}{2}B = \frac{(\lambda - A)v_x}{v_y};$$

$$\frac{(\lambda - A)v_x}{v_y}v_x = (\lambda - C)v_y; \quad => \quad \left(\frac{v_x}{v_y}\right)^2 = \frac{\lambda - C}{\lambda - A}.$$

Hence $\theta = \tan^{-1}\left(\sqrt{\frac{\lambda - C}{\lambda - A}}\right)$.

In the second solution, the declination angle will not have a sign (i.e., it will always be between 0º and 90º), so the sense of rotation must be obtained from the geodesic torsion. Care must be taken, since the sign of geodesic torsion is also affected by whether the surface is convex or concave.

It is preferable to measure rotation of the minor axis, as it corresponds to the longer curvature radius, and therefore shows elongation direction of the feature (i.e., describes the orientations of troughs and ridges).

Paraboloid flattening

We can estimate how meaningful the results are by calculating the flattening of the osculating paraboloid, which can be done by comparing the radii of osculating circles of parabola traces on planes of principal curvatures. To indicate flattening, an angle between the major radius and a normal to a tangent to the circle of a minor radius was trigonometrically determined: $\alpha = \cos^{-1}\frac{b}{a}$ (Fig. 31a). An angle was chosen over a ratio so that the output may be used in the creation of Second Derivative AVA (Fig. 31b).

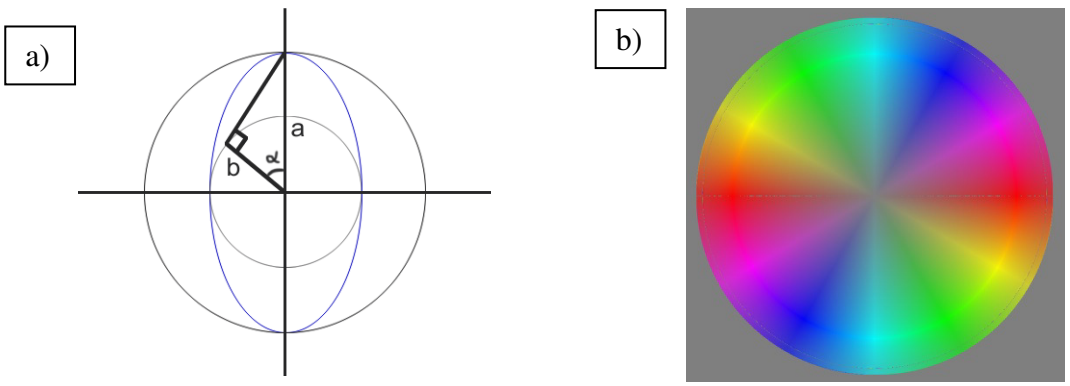


Figure 31: a) Finding the degree of flattening, b) Second derivative AVA obtained from the surface of a concave half-sphere, with 11px kernel radius.

Hessian Matrix

In addition, wherever the surface first derivative is zero, the Hessian determinant may be used in conjunction with $\frac{d^2z}{dx^2}$ to check if the points are local minima, maxima or saddle [Anton and Rorres, 2010]. For second order derivatives the Hessian matrix is $H(x,y) = \begin{bmatrix} A & B \\ B & C \end{bmatrix}$.

Its determinant is $\det[H(x,y)] = A * C - (B)^2$.

Visualization

The degree of minor axis rotation and the degree of flattening can be used as inputs for the AVA tool. This generates visualisations of the second derivative that are similar to those of Strike and Dip for the first derivative. Since only a 180° variation exists, it is reasonable to multiply the rotation value by 2 before feeding it to AVA or constructing histograms, such that the entire colour wheel is included and representation is unaffected by the directional sense [Davis, 2002].

Conceptual model

To summarize, computing the Second Derivative AVA begins with running a custom *Convolution* script with kernels provided by the *Make Convolution Kernel* script to obtain curvatures and geodesic torsion. All intermediate steps are computed via ArcGIS Model. Visualization of Curvature is then produced by running the *AVA script tool* with doubled *Minor Axis Degree of Rotation* in place of *Strike* and with *Paraboloid Flattening Angle* in place of *Dip*. (Fig. 32)

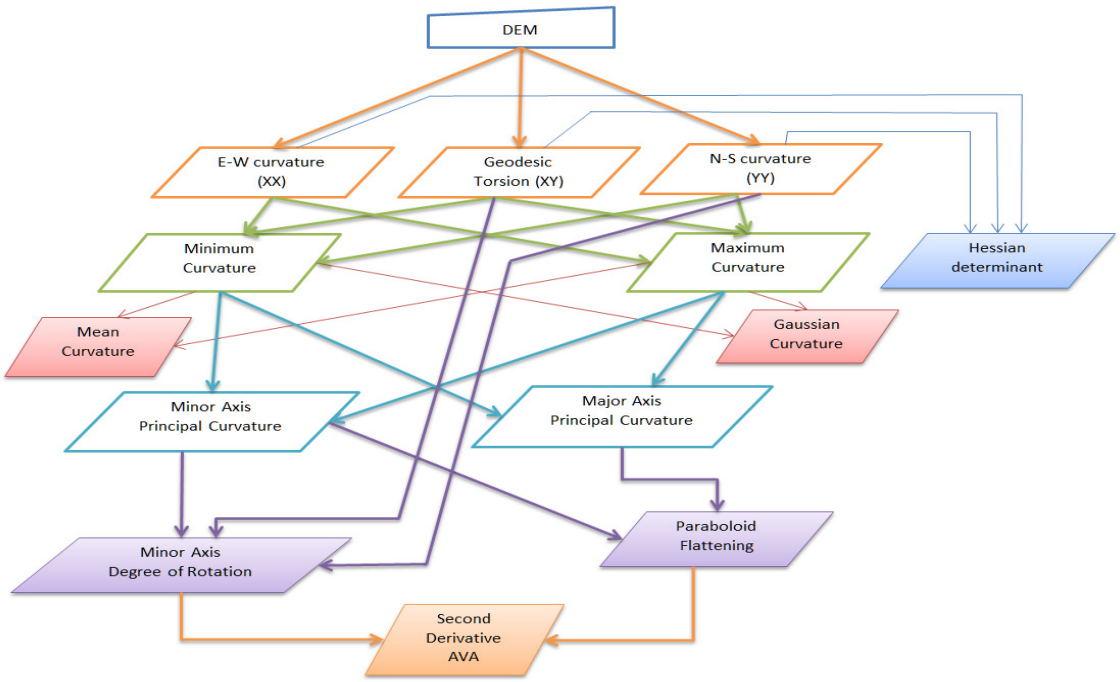


Figure 32: Conceptual model for computing Second Derivative AVA

METHODOLOGY AND IMPLEMENTATION

Standard basis curvature and geodesic torsion

A second order polynomial $Z = Ax^2 + Bxy + Cy^2 + Dx + Ey + F$ completely describes the first and second derivatives. For instance, principal curvatures are given by parameters *A* and *C*, from which Mean Curvature and Gaussian Curvature can be readily found. The mixed derivative is given by parameter *B*. Since *F*=*Z* at the middle of the kernel, this parameter of the polynomial approximation is identical to the DEM values.

The convolution kernel for parameter A is found by setting it to one, and all other parameters to zero, hence simplifying the polynomial to $Z = x^2$, which is then used to calculate elements of a convolution matrix. Weights are then applied to create the circular kernel. To ensure that the convolution matrix adds up to zero the mean value of the circular kernel is then subtracted from each element in the array in a square kernel, and

weights are applied again to make that kernel circular. The mean value for circular kernel is found as a ratio of the sum of all elements divided by the area of a circle. The kernel is then multiplied by a normalization factor, to obtain the final convolution matrix.

In short, convolution matrices for calculating parameters for this polynomial can be obtained by solving for

$$G_{xx} = \frac{g_A - \overline{x^2}}{\sum(x^4) - \sum(x^2y^2)}\ ,\ G_{yy} = \frac{g_C - \overline{y^2}}{\sum(x^4) - \sum(x^2y^2)}\ ,G_{xy} = \frac{g_B}{\sum(x^2y^2)}\ .$$

The kernels can be computed in Excel, however that process is tedious, time consuming and prone to human error due to large number of steps involved. The Create Convolution Matrix script tool [source code in Appx. 4] generates the appropriate, normalized circular kernels for curvatures and for geodesic torsion.

Below is the pseudocode for the process, where uppercase letters represent matrices, lower case represent numeric variables, $J \circ K$ represents Hadamard product between matrices J and K, $\Sigma(J)$ represents sum of all elements in J.

For curvature kernels (G_{xx} and G_{yy}):

1) Compute the base kernel, e.g. $Q_A(x,y)=x^2$, $Q_B(x,y)=x*y$, $Q_C(x,y)=y^2$ where Q is a kernel matrix and x and y are coordinates, in pixels, from the kernel centre
2) Compute the circular kernel weight matrix, W, following procedure outlined in Chapter 3
3) Determine vertical offset, $v = \Sigma(Q \circ W)/\Sigma(W)$
4) Subtract vertical offset from every element in the base kernel, $T = Q - v$
5) Compute sums $q_{aa} = \Sigma((Q_A \circ W) \circ (Q_A \circ W))$ and $q_{bb} = \Sigma((Q_B \circ W) \circ (Q_B \circ W))$
6) Determine normalizing coefficient $k = (q_{aa} - q_{bb})$
7) With grid spacing s_x, s_y, normalize the kernel: $G = (T \circ W) / (s_x * s_y * k)$

For geodesic torsion kernels:

1) Compute the base kernel, e.g. $Q_B(x,y)=x*y$ where Q is a kernel matrix and x and y are coordinates, in pixels, from the kernel centre
2) Compute the circular kernel weight matrix, W, following procedure outlined in Chapter 3
3) Compute sum $q_{xy}= \Sigma((Q_B \circ W) \circ (Q_B \circ W))$
4) With grid spacing s_x, s_y, normalize the kernel: $G_{xy} = (Q_B \circ W) / (s_x * s_y * k)$

In this chapter, the results of filtering the DEM with various convolution kernels for second derivative were explored. Grid spacing is assumed uniform in all these calculations. Kernels produced by the Create Convolution Matrix script tool were used as inputs for the Custom Convolution tool described in the previous chapter. The tools were combined into a model (Fig. 33).

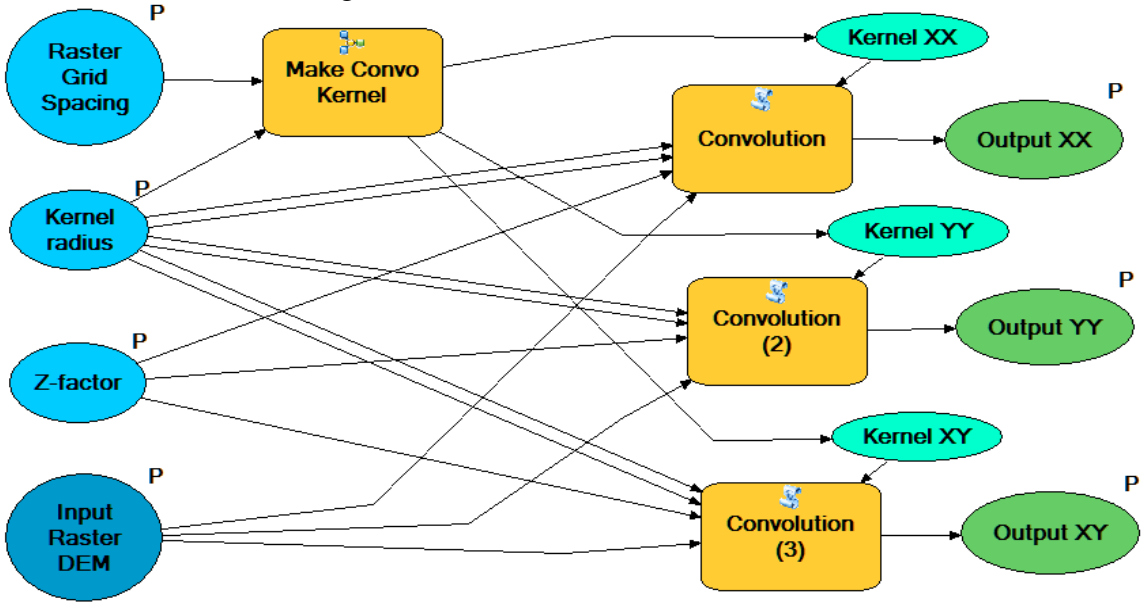


Figure 33: An ArcGIS model utilizing *Make Convolution Kernel* script tool to provide *Convolution* script tool correct strings of characters to conduct convolution with a properly normalized circular kernel.

Post-processing:

All post-processing raster algebra has been conducted using the *Spatial Analyst* extension – *Math* toolset geoprocessing tools, sequenced in *ArcGIS ModelBuilder* as *ArcGIS model*. (Fig. 34)

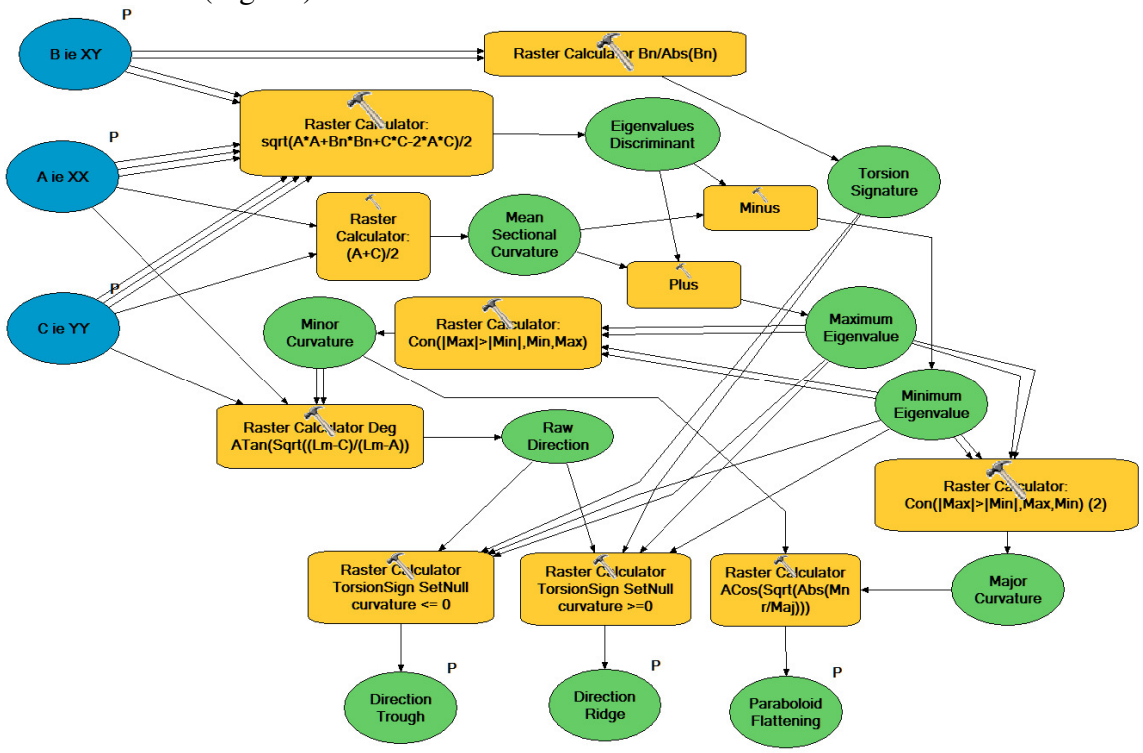


Figure 34: "Calculate direction", the ArcGIS model to generate rasters for the trends of troughs, ridges, and to compute Paraboloid Flattening

Calculation of the Second Derivative AVA takes curvature outputs from the convolution to produce the degree of short-axis rotation and parabolic flattening, which are then supplied to the AVA script as inputs. Variables *A*, *B*, and *C* are parameters of the polynomial, *A* corresponding to curvature *XX*, *B* – to geodesic torsion *XY*, and *C* corresponds to curvature *YY*. The tools used in the models are from the Map Algebra

toolset in the Spatial Analyst toolbox, and hence depend upon the Spatial Analyst extension to be enabled.

Intermediate data is not normally saved, but could be saved by modifying the model by changing the properties of a *Data Variable* from *Intermediate* to *Model Parameter* (Fig. 35).

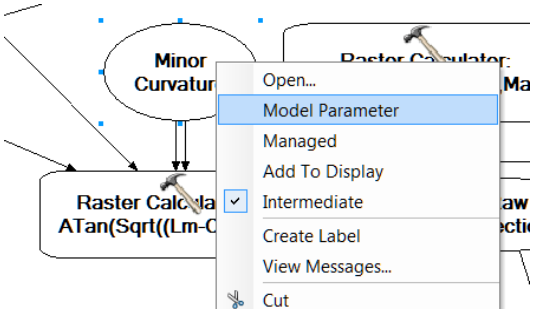


Figure 35: Saving intermediate data is possible by selecting an intermediate checkmark in the modelbuilder context menu for a variable.

Minor and Major Curvatures are found from the Eigenvalues by using Map algebra expressions identical in all but the order of inputs in the conditional statement.

The Hessian is not used in making of Second Derivative AVA, so no model for it was created. The Hessian was instead computed in Raster Calculator using the following formula: *(XXcurvature.tif)*(YYcurvature.tif)-(XYcurvature.tif)* (XYcurvature.tif)*

Third derivative

Furthermore, as was mentioned earlier, it is possible to make even higher derivatives, by using any of the second derivative outputs as inputs for the first derivative tool in a manner similar to a regular DEM. For instance, *Mean Curvature* can be supplied to the *All-In-One* tool as input. Using the convolution technique to calculate the third derivative directly is not as straightforward as for the curvature and slope, since higher

derivative convolution matrices are not always independent from the lower ones. Convolution matrices can be checked for dependence by taking the Hadamard product of the two. If the sum of elements in the resultant matrix is other than zero, these matrices are not independent. For instance, convolving a higher order matrix, such as for X^2Y with a matrix for Y will yield non-zero result.

Third derivative can be obtained simply by taking a first derivative of a second derivative (i.e., running an All-In-One script described in chapter 3), or the First Derivative tool described in chapter 2 over one of the curvature rasters (for instance, Mean Curvature raster). The quality of the third derivative depends on the texture of the second derivative. If the second derivative has many short wavelength features, then the third derivative will be noisy. When running these tools, AVA z-factor should be adjusted to compensate for the lower amplitude of a signal, i.e. lower rates of change. Since the z-factor needed to ensure the widest range of saturation values on the third order AVA will greatly depend upon surface roughness and kernel size, no analytical recommendations are given as to the best choice of such; the appropriate values can be found through trial-and-error.

For consistency, all images in this chapter are derived from the same dataset of a Hebes Chasma mosaic at 50 m/px spatial resolution. This was chosen in place of the test image used earlier, such that more regional trends could be observed.

RESULTS

This process was tested on Hebes Chasma, Mars, mosaic DEM at 50 m/px resolution. Tested were square kernels vs. circular and recursion vs. convolution. Intermediate datasets were retained and provided for demonstration purposes in this section. Additionally, applications for feature registration were explored.

Convolution

Examining the outputs from recursive convolution demonstrated that the *x*-gradient of the *y*-gradient ($\frac{d}{dx}\left(\frac{dz}{dy}\right)$) produced the same output as the *y*-gradient of the *x*-gradient ($\frac{d}{dy}\left(\frac{dz}{dx}\right)$), which is consistent with Schwarz's theorem on equality of mixed partials. Furthermore, the mixed partial derivative was visually similar to Parameter *F* from convolution in *XY*. Likewise, partial derivatives $\frac{d^2z}{dx^2}$ and $\frac{d^2z}{dy^2}$ were similar to Parameters *D* from X^2 and *E* from Y^2, respectively (Fig. 36).

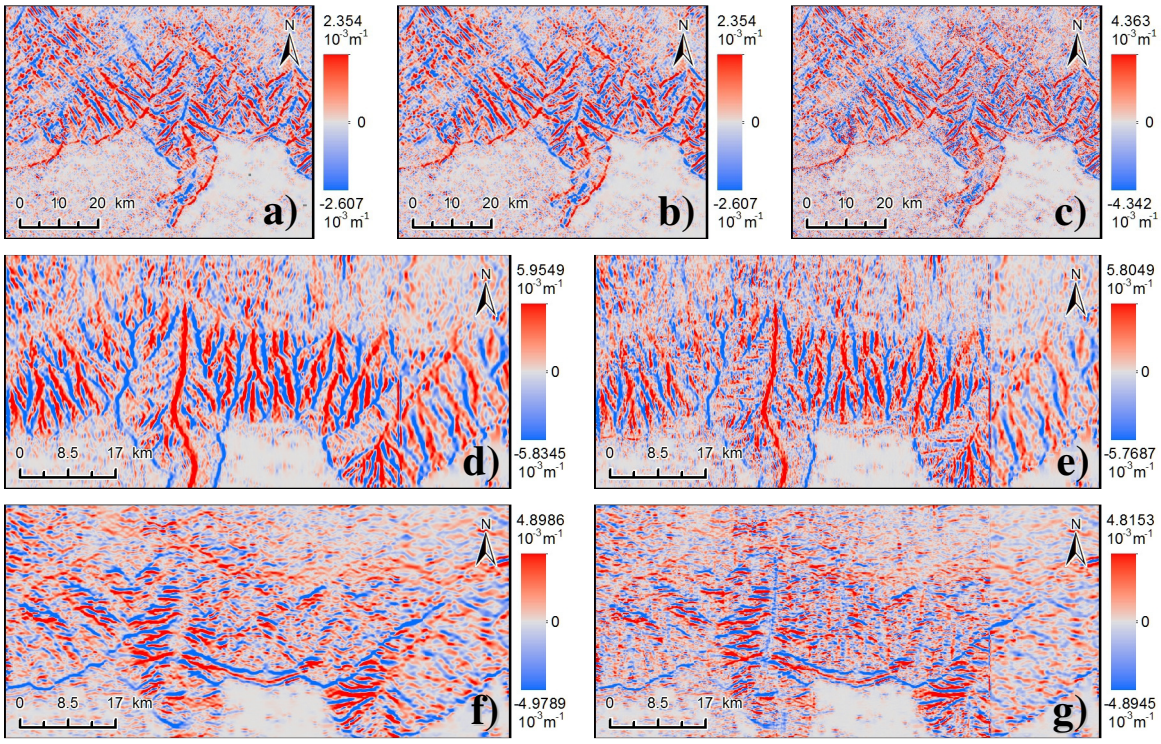


Figure 36: Comparison of second derivatives obtained by recursive convolution and direct convolution, conducted using circular kernel with 5 px radius, displayed using a colour ramp shown on the left, using a ArcGIS Standard Deviation stretch type at n=2.5 setting.
a) Y-gradient of x-gradient, b) x-gradient of y-gradient, c) mixed partial obtained directly; d) X gradient of X gradient, e) X curvature, f) Y gradient of Y gradient, g) Y curvature.

Every parameter of 2nd order polynomial was then calculated directly by convolution (Fig. 37).

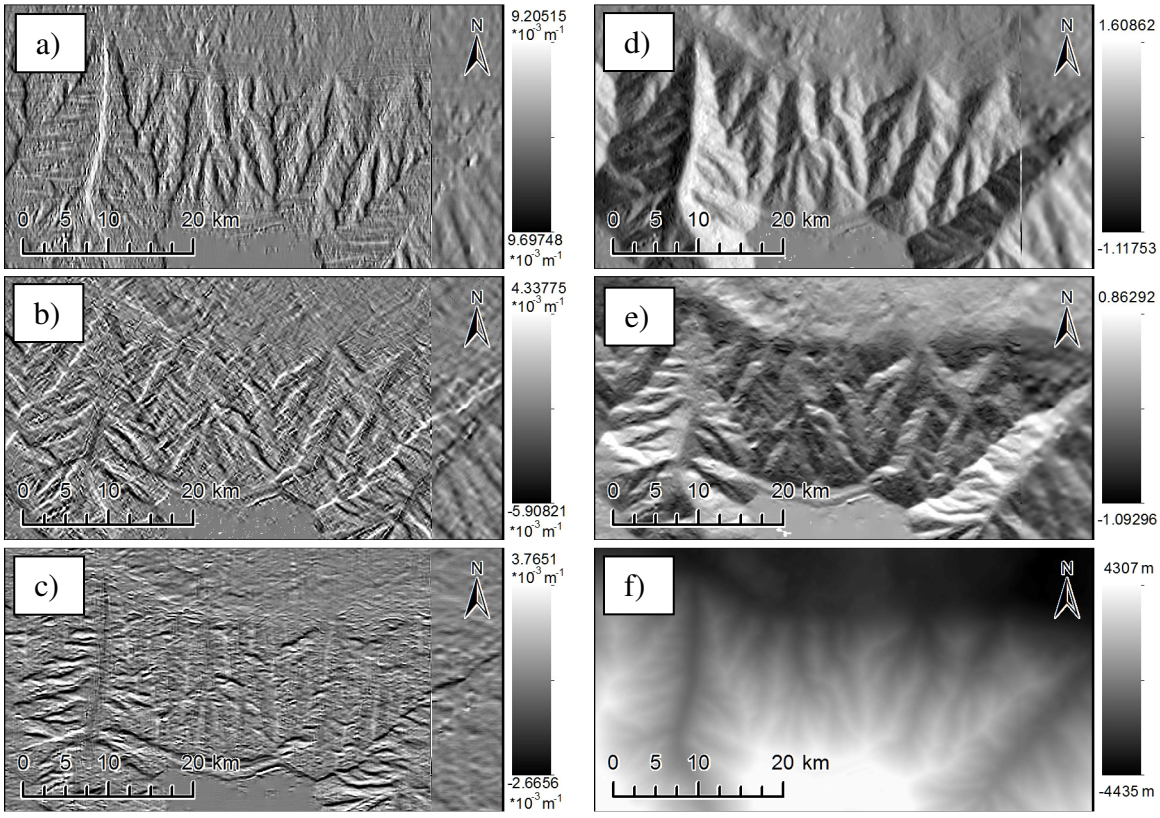


Figure 37: Convolution outputs obtained with circular kernel of 2 px radius provide parameters for the equation $Z = Ax^2 + Bxy + Cy^2 + Dx + Ey + F$ a) Parameter A, from X^2, b) Parameter B, from XY, c) Parameter C, from Y^2, d) Parameter D, from X, e) Parameter E, from Y, f) Parameter F, source DEM.

Post-processing

Hessian determinant

Convolution provides standard basis curvatures and geodesic torsion. On its own, such data is not very useful, and further post-processing is required to obtain meaningful information. The Hessian determinant is valuable for identifying topographic anomalies such as faults, channels and ridges, which manifest as lineations (Fig. 38).

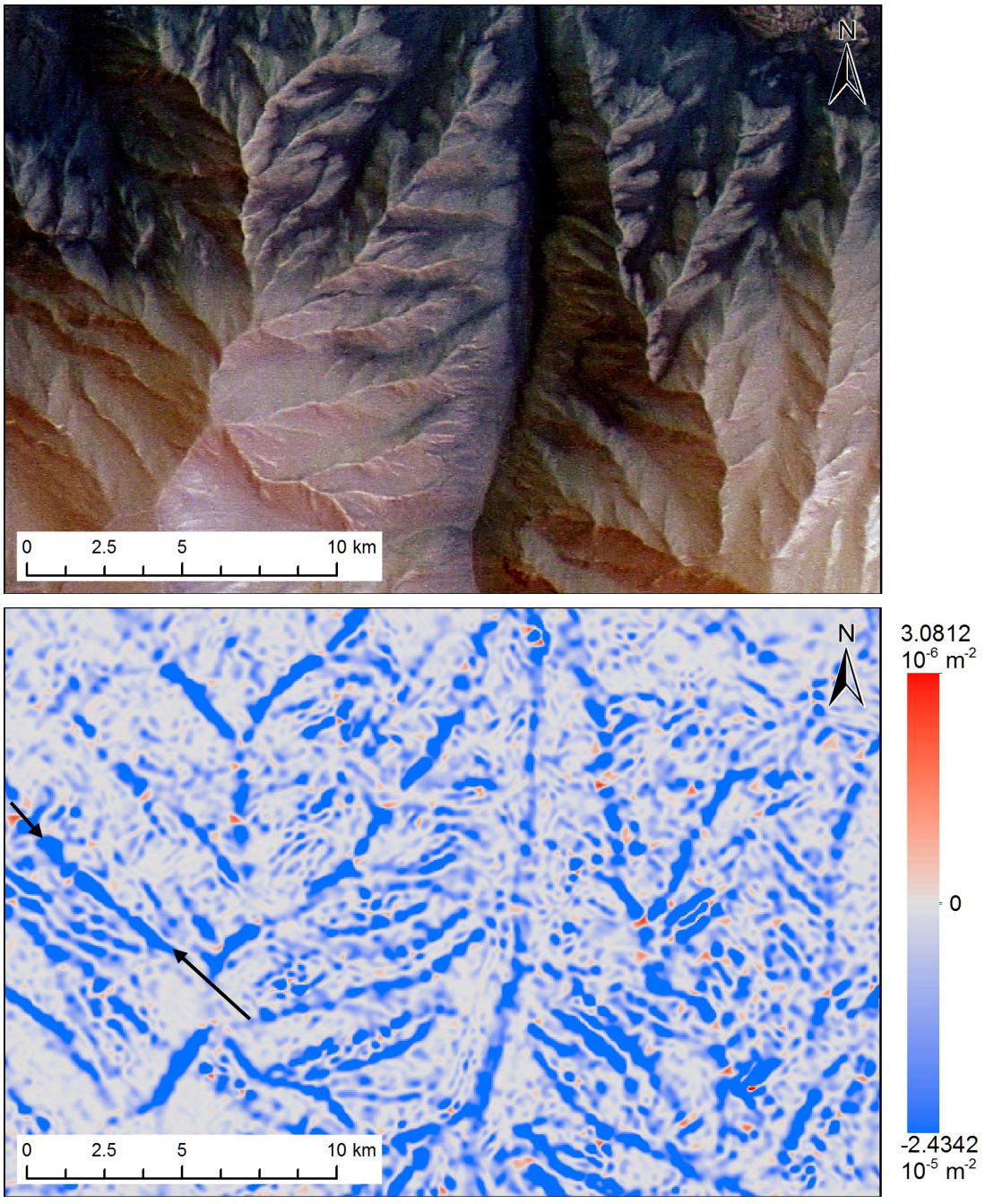


Figure 38: Hessian determinant (bottom) and HRSC colour mosaic [Appx. 1] (top). On the Hessian, positive values represent relative extrema (peaks and sinks) and negative values represent saddle points. Arrows point to a lineation. Hessian determinant was calculated from curvatures obtained using a kernel with radius of 5px.

Principal curvature

Eigenvalues of a quadratic polynomial, or principal curvatures, were found to be useful on their own, not only as an intermediate step for further processing. For instance, as is, λ_{min} is better suited for ridges and faults examination, while λ_{max} is more fitting for work in erosional features, such as fluvial geomorphology, for instance estimating the channel thalweg (Fig. 39).

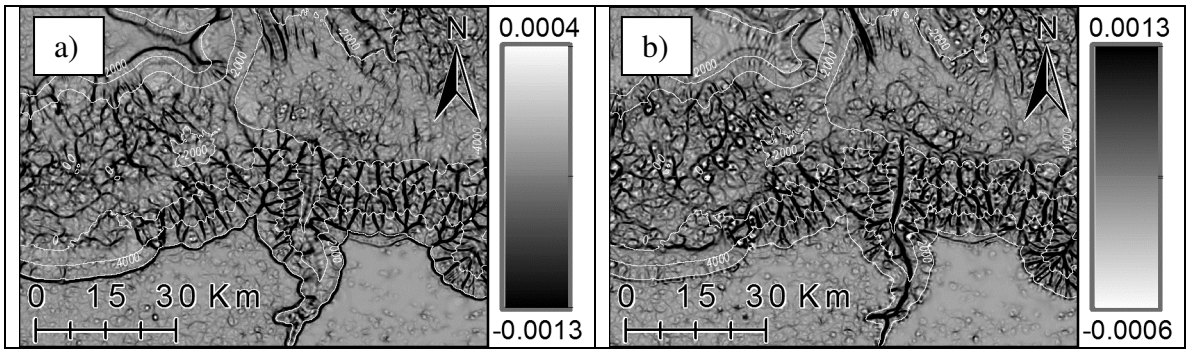


Figure 39: Eigenvalue rasters obtained with 5x5 convolution kernel:
a) smaller eigenvalue, λ_{min}, is best suitable for identifying ridges;
b) larger eigenvalue, λ_{max}, best displays concave features, making it suitable for tasks such as locating the channel thalweg.
Units are m^{-1}. For clarity, images are overlaid with elevation contours (in white) plotted at 1000 m intervals.

They also provide a starting point for further study. From these, Mean and Gaussian curvatures have been found (Fig. 40 a, b). Comparing principal curvature absolute magnitudes allowed for extraction of Major and Minor curvatures (Fig. 40 c, d), from which Angle of Rotation of the Minor Axis (Fig. 40 e), and Paraboloid Flattening (Fig. 40 f) were found.

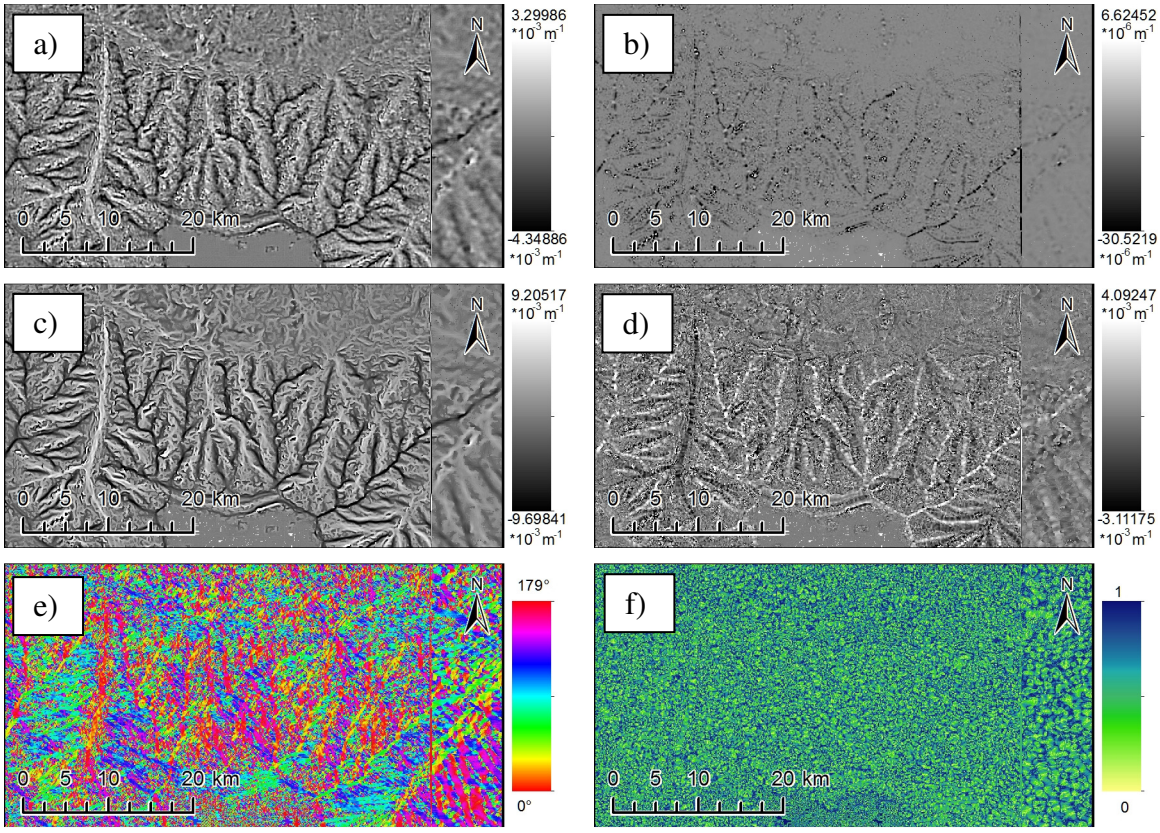


Figure 40: From 5x5 convolution a) Mean Curvature, b) Gaussian Curvature, c) Long Axis Principal Curvature, d) Short Axis Principal Curvature, e) Angle of Rotation of the Short Axis, and f) Paraboloid Flattening ratio.

Histograms

The curvature orientation was computed for Hebes 50m/px HRSC mosaic dataset using an 11 px radius circular kernel. The frequencies of occurrences of different orientations were plotted onto rose diagrams. The result showed unexpected peaks at 0°, 90°, 180° and 270° (Fig. 41a). Rotating the dataset rotated these unexpected peaks (Fig. 41b), indicating that they are the result of artifacts inherited from input DEM.

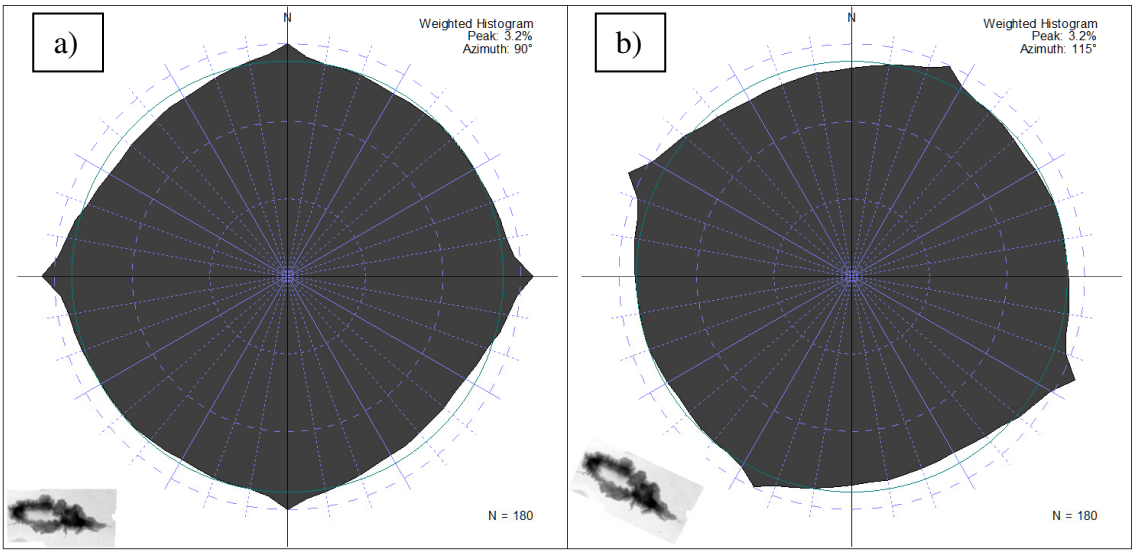


Figure 41: Frequency of occurrence of pixels with each particular orientation value plotted in SpheriStat (© Pangaea Scientific) as trend axial data peaks histogram. Class alignment centered at 0°N. Thumbprint for each dataset in the bottom left corner.
a) Curvature orientation for 50m/px Hebes mosaic dataset, processed with 11 px radius curvature convolution kernels.
b) Same dataset rotated 25° and processed in the same way, demonstrating that the frequency peak anomaly on the histogram rotated with the dataset.

The Hebes HRSC mosaic was then clipped to a polygon outlining the chasma and processed with a small (2 px radius) and large (11 px radius) kernel (Fig. 42). The unexpected peaks seen in the histogram for the entire dataset are no longer apparent on the clipped raster, indicating that the anomaly was primarily pertaining to the flat regions of the plateau. The smaller kernel shows peaks at 0°, 30°, 45°, and 60°. These peaks are not present on a frequency plot of curvature raster produced with a larger kernel. Hence, they are likely due to pixilation effects. This indicates that convolution outputs for curvature only become meaningful if the kernel is sufficiently large.

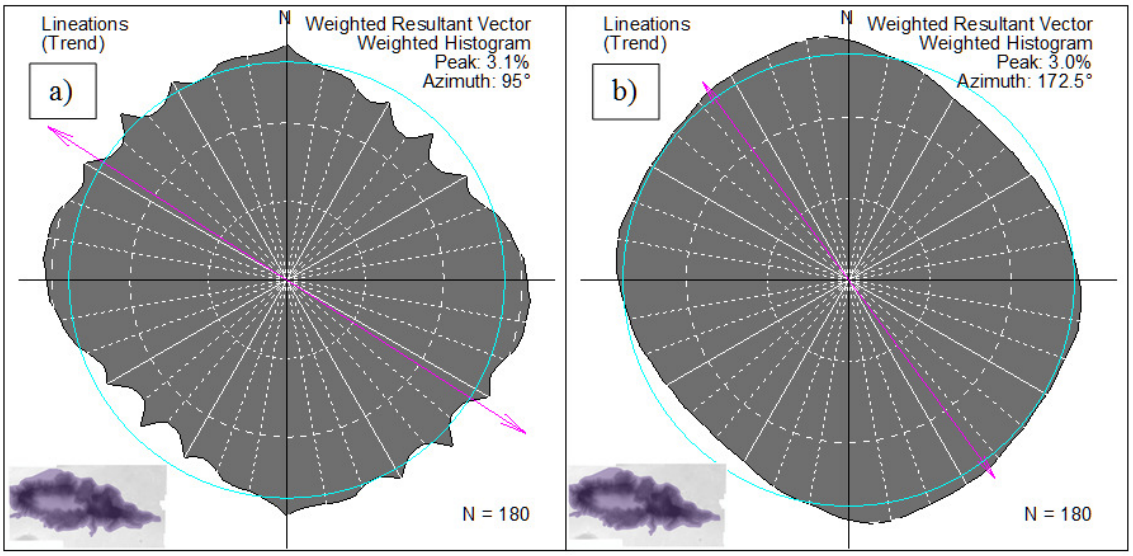


Figure 42: Frequency of occurrence of pixels with each particular orientation value plotted in SpheriStat (© Pangaea Scientific) as trend axial data peaks histogram. Class alignment centered at 0°N. Polygon used for clipping the dataset is shown in the thumbprint image in the bottom left corner. a) Curvature orientation for 50m/px Hebes mosaic dataset, thumbprint in bottom left corner, processed with 11 px radius curvature convolution kernels. b) Same dataset rotated 25° and processed in the same way, demonstrating that the frequency peak anomaly on the histogram rotated with the dataset.

Subsequently, the same dataset was processed using circular kernels of 2 px radius using convolution and recursion methods. The output data from two methods was compared by means of a heat map (Fig. 43). It was thus discovered that outputs of recursion showed bias towards principal directions of 0°, 90°, 180°, and 270° as compared to direct convolution outputs. Determining the cause of the discrepancy is beyond the scope of this thesis. Direct convolution for curvature was concluded to yield better results than recursive convolution for gradient, so it was used henceforth.

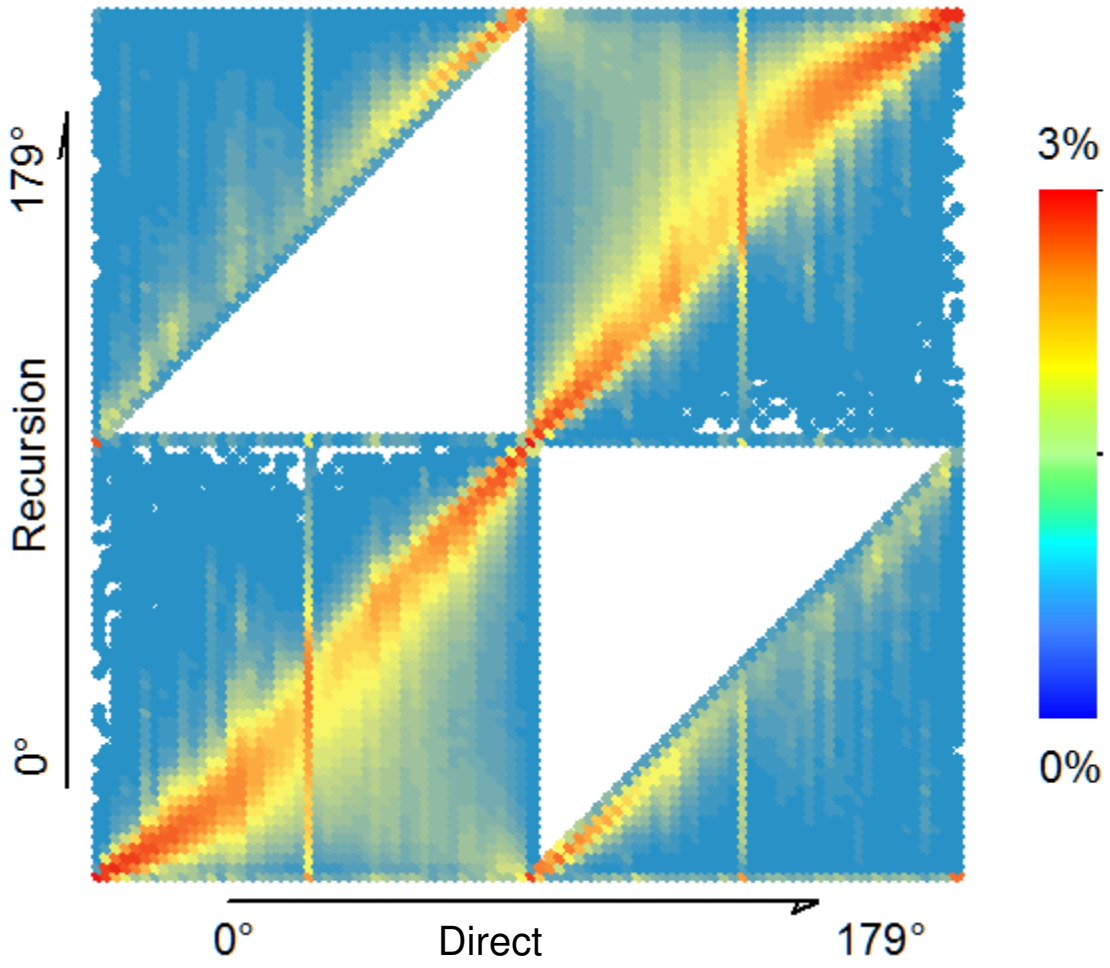


Figure 43: Heat map of the frequency of occurrence of each particular combination of curvature orientations pooled at 2° intervals from direct convolution and recursive convolution techniques run using kernels with a 2 px radius.

Second derivative AVA

The second derivative AVA displays the asymmetry in curving of the surface, indicating the direction of lesser curvature, i.e. elongated features (Fig. 44). It is produced from the orientation of the paraboloid's long axis and its flattening.

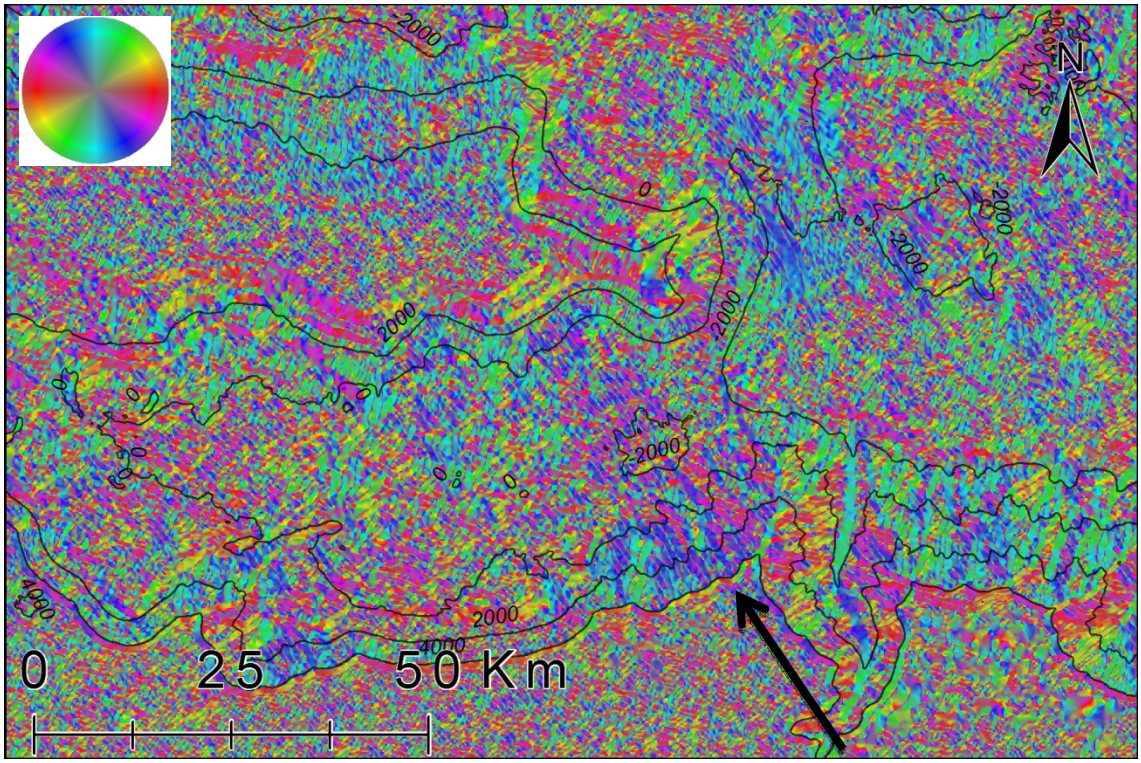


Figure 44: Second derivative AVA, of the same DEM, obtained with 5x5 kernel, showing the reference colour wheel in the top left (produced as second derivative AVA of a concave half-sphere). Large black arrow on the bottom right points at a set of lineations traceable across the ILD. For clarity, image is overlaid by elevation contours drawn at 1000 m intervals.

Since a paraboloid is symmetric also along the minor axis, the major axis direction can be defined completely by a range of values from 0° to 180°. To ensure that the visualization spans the entire colour wheel, the declination angle was doubled prior to rounding to the nearest integer, making even very subtle features readily apparent.

Third derivative AVA

The third derivative AVA can be useful for such tasks as highlighting the crests. It can be found by running a first derivative tool over the curvature raster. The most effective results were obtained when the orientation was sourced from the minimum

curvature raster, and saturation was computed from the mean curvature raster. Since magnitude of the derivative signal is different from that of the topography (usually much weaker), it is necessary to apply vertical exaggeration to ensure further processing with the First Derivative tool produces a vividly coloured AVA. Vertical exaggeration was handpicked by trial and error to provide the widest sweep across present values (Fig. 45).

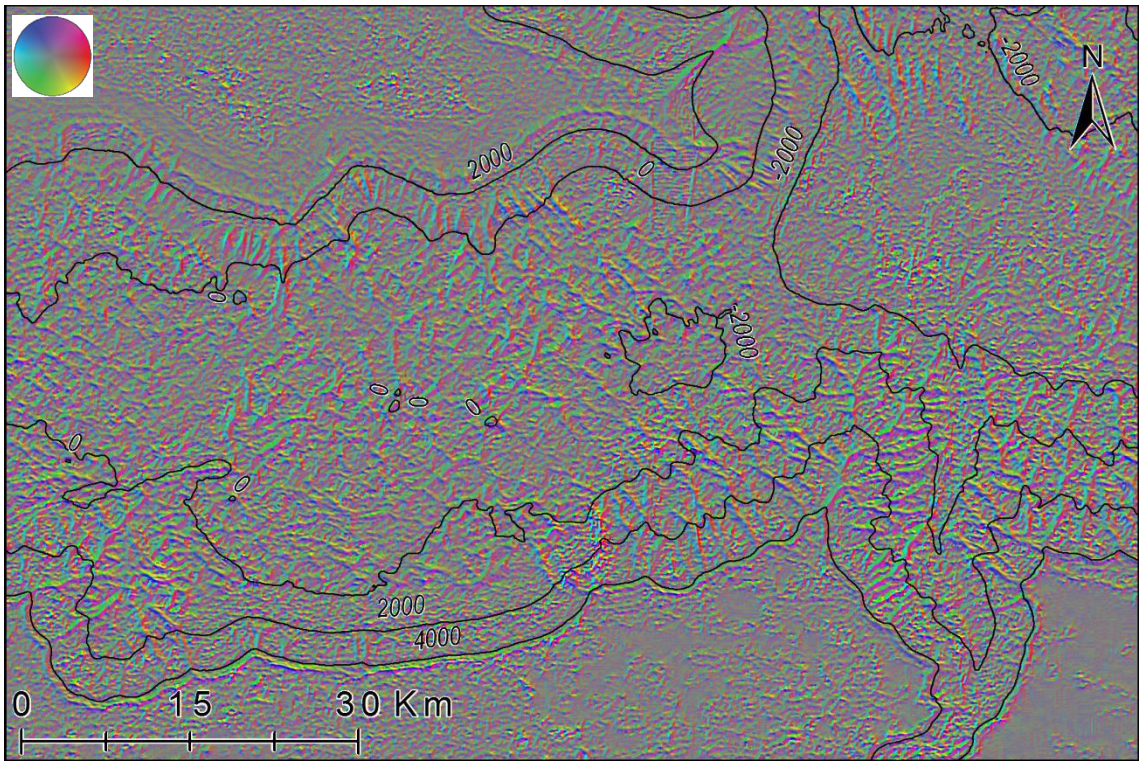


Figure 45: Third derivative AVA, made using 2 px radius maximum curvature's 2 px radius gradients. Default AVA colour map is applied, shown in the top left corner.

Feature registration

An important consequence of being able to obtain higher order derivatives is in the fact that it opens way to feature registration and segmentation, i.e. extract features in the vector format (e.g. polylines, polygons). Features in vector format can be extracted

and analyzed for quantitative properties with other tools readily available in GIS programs. For instance, the second derivative direction raster can be further processed for circular variance to locate boundary values, by applying a threshold to segment it into regions with consistent orientation. Circular variance [Davis, 2002] can be found as $S^2 = \frac{n - \bar{r}}{n}$; where $\bar{r} = \sqrt{(\overline{\cos\theta})^2 + (\overline{\sin\theta})^2}$

By means of the ArcGIS Thin tool in Spatial Analyst extension, these regions can then be converted into a vector polyline format (Fig. 46).

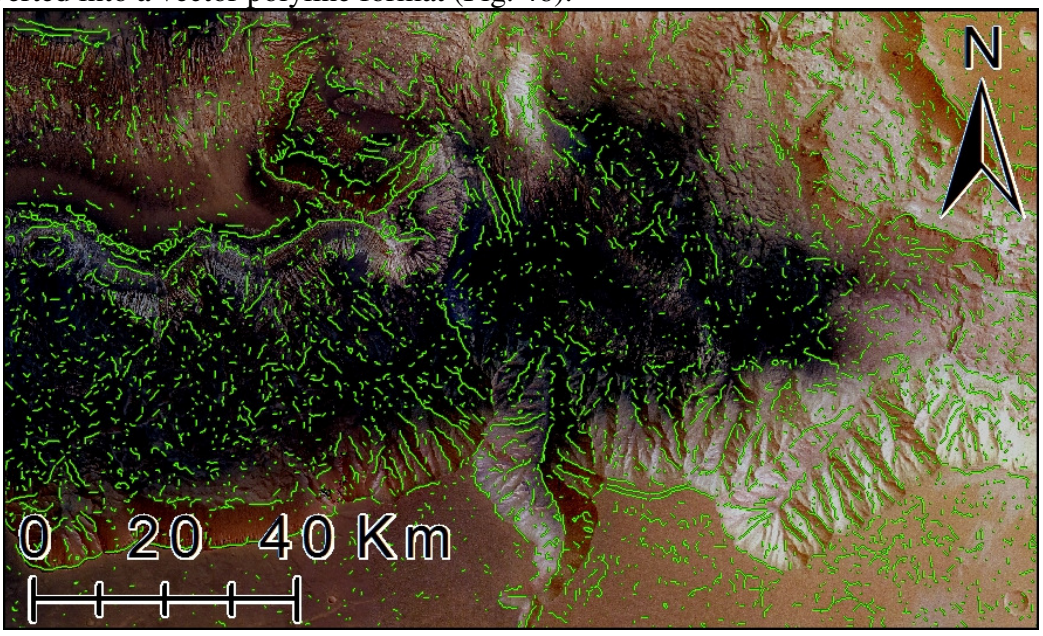


Figure 46: Lineations registered by computer overlain on the false-colour raster image.

Orientations of the lines were calculated in ArcGIS field calculator using a python expression [https://geonet.esri.com/thread/27393]

*Int(180 + math.atan2((!Shape.lastpoint.X! - !Shape.firstpoint.X!),(!Shape.lastpoint.Y! - !Shape.firstpoint.Y!)) * (180 / math.pi)+0.5)%180*

Taking a mode of 180 collapses the distribution into a range of 0-180, which allows lineation orientation to be represented by ArcGIS symbology options. (Fig. 47)

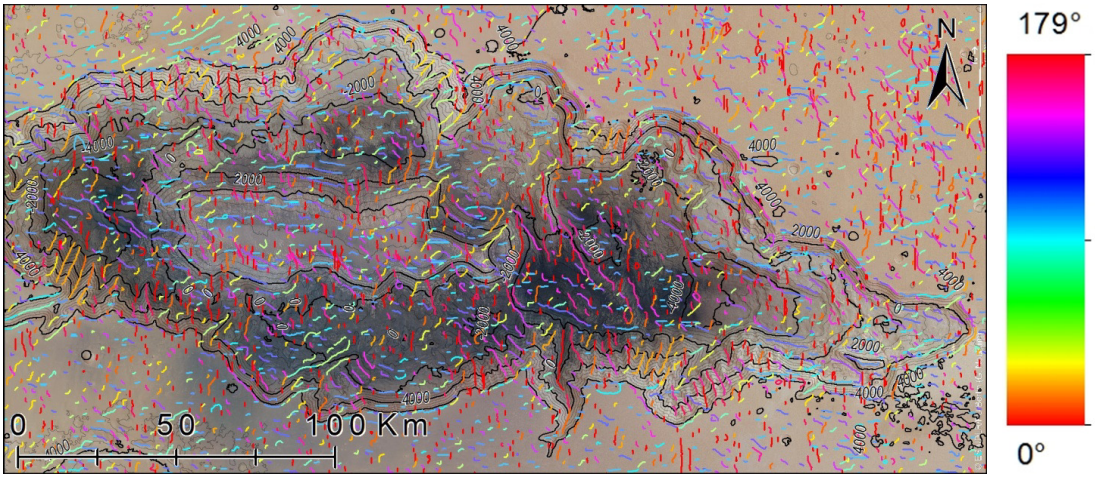


Figure 47: Lineations registered by computer by skeletonization of regions with consistent orientation, displayed over on the RGB HRSC mosaic. The attitude of the lineations is represented by colour ramp on the right.

The lengths were plotted in SpheriStat (© Pangaea Scientific) to obtain frequency distribution plot (Fig. 48).

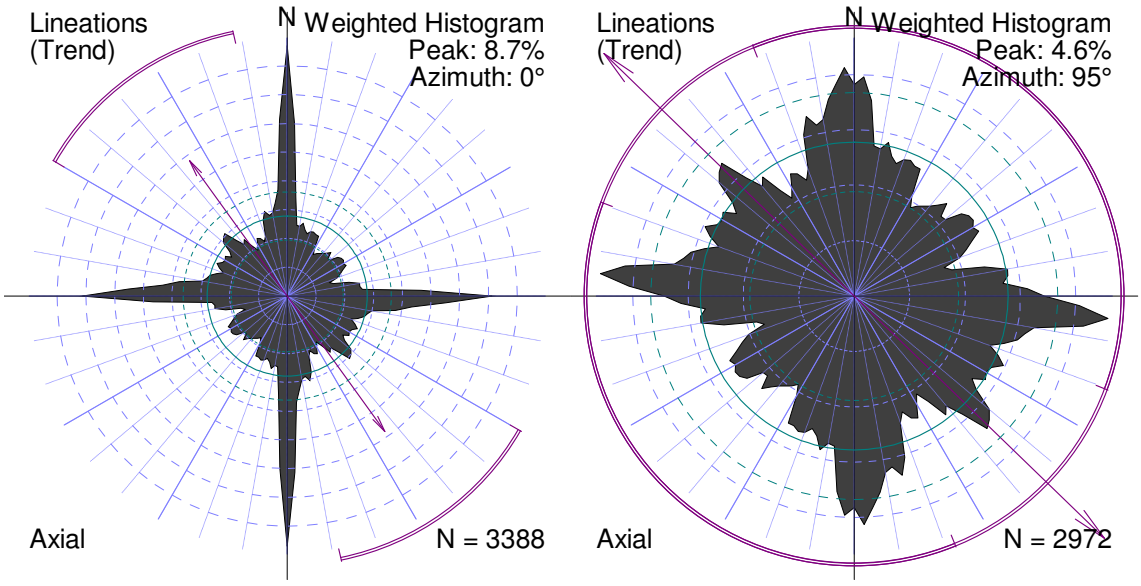


Figure 48: Frequency distribution plot of lineations, weighted by lineation length. Top: scatter plot of cumulative lineation length per azimuth. Bottom left: as a rose diagram, produced with SpheriStat (© Pangaea Scientific). Bottom right: same, but with features trending at 0, 90 and 180 removed from the dataset.

DISCUSSION

Tools provided by the Spatial Analyst Extension toolkit are insufficient for calculation of principal curvatures, because they lack the ability of computing geodesic torsion or allow for a choice of kernel size. Although this work employs Spatial Analyst Extension for Map Algebra tasks due to its convenience, a Spatial Analyst Extension is not essential, as it can be readily substituted by any other software capable of performing algebraic operations on raster datasets.

Two approaches for calculating the second derivative were investigated: through recursive gradient convolution, and through direct convolution to curvature. While producing similar results, the first method required double the processing time, while also doubling the cropping margins of the input raster. The recursive approach was also found to be inferior due to having greater susceptibility to arithmetic error and noise.

While Mean and Gaussian Curvature are of significant importance to differential calculus of embedded manifolds, they appear to be only of marginal use in geological studies of topography. Whereas Mean curvature can be used as an intermediate step towards the generation of third derivative AVA, there are no obvious uses for Gaussian curvature.

The second derivative AVA effectively highlights topographic signatures of tectonic structures, possibly faults. Second derivative orientation data can be used for segmentation and registration of lineations, allowing quantitative data to be extracted and analyzed.

The third derivative AVA representation was found to be effective in highlighting orientations of ridges and troughs, with hue corresponding to the direction of lowest gradient, i.e. displaying colour opposite from the first derivative AVA.

As will be demonstrated in the following chapter, the second derivative AVA on its own was not sufficient to decide whether the feature was extensional or compressional as it only effectively displays the trend of the feature. Nevertheless, Second derivative AVA of curvature is effective means of tracking tectonic signatures, even on a relatively flat plateau. On spur-and-gully textures, using a third derivative of short axis principal curvature may be easier, as AVA of curvature is very sensitive.

The higher order derivatives allow for identification of linear features that appear to trace continuously from the walls of Hebes Chasma into the ILD (Fig. 43), features invisible in DTM, raster imagery, or even in First Derivative AVA [Schmidt, 2015].

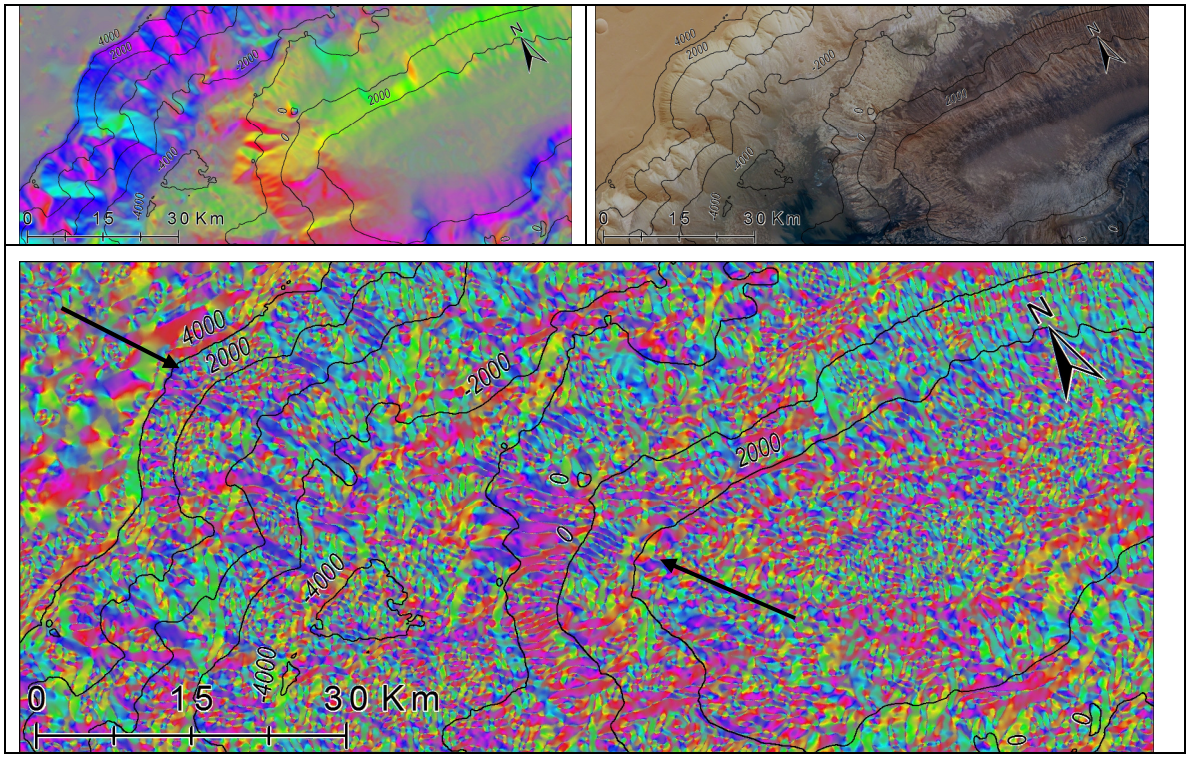


Figure 49: Bottom: Second derivative AVA produced with 5x5 kernel overlaid by contour lines, showing multiple lineations (purple/blue) extending from the plateau into the ILD. Black arrows point to a lineation. First derivative AVA on the top left and satellite imagery on the top right are provided for comparison.

The computing of convolution is rapid, even for large kernels, e.g. running Convolution script with 11x11 kernel over 1516x875 image three times took only a total of 20 seconds. However, the post-processing tasks conducted by ArcToolbox Models took considerably longer, as they involve large number of geoprocessing operations:
1 minute and 20 seconds to convert from directional curvatures and geodesic torsion to Second Derivative orientation, and an additional 40 seconds to generate the AVA.

Registration of lineations from curvature orientations by examining circular variance was successfully accomplished, however the distribution of output polyline

orientations showed unnatural peaks at 0° and 90°, indicating that information was distorted in the process and the procedure needs further study. The most likely cause of distortion is rounding and generation of angularity due to pixilation effects during binary raster region skeletonization.

CONCLUSIONS

By comparing the results, it was concluded that the recursive convolution for gradient and direct convolution for curvature produced identical results, and hence, the methodology of direct convolution to the second derivative was proven correct.

The second derivative AVA, effectively highlights topographic signatures produced by the declination of principal axes and their ratio to one another. These topographic features may aid in identifying and interpreting geological structures.

Chapter 5: Geological Applications – Case Studies

INTRODUCTION

In this chapter, tools developed in this thesis are applied to several geological problems, primarily focused on the Tharsis rise, a large Martian volcano-tectonic province (Fig. 50).

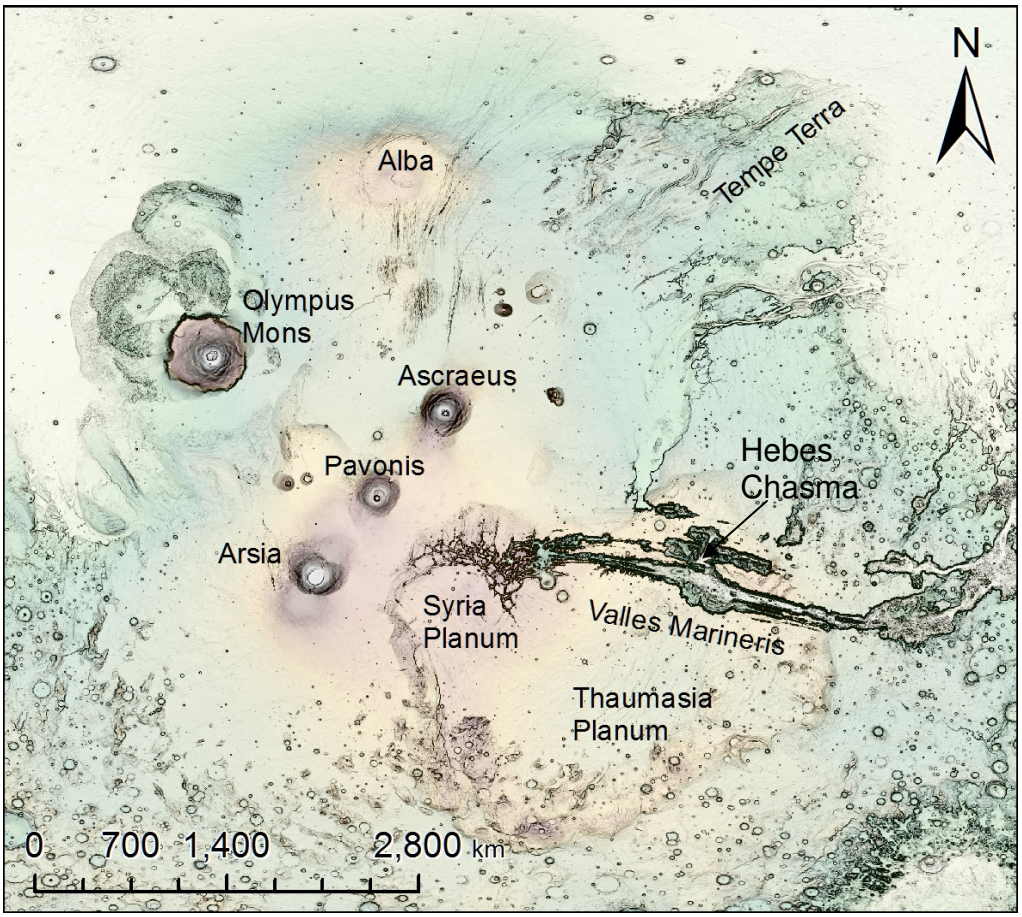


Figure 50: Tharsis bulge as seen on RGB representation of MOLA DEM with high areas in red and low areas in white [PlanetServer, 2015], overlaid by a vector sum of gradients derivative raster (the square root of the sum of squares of gradients, i.e., slope). Basemap colour key is unavailable.

Geologic Background – Tharsis history:

Mars is a terrestrial planet that has experienced a rich geologic history, with limited erosion and no plate tectonics. It has thus preserved much of its early geological records. The appearance of Mars is dominated by a topographic dichotomy with a north hemisphere topographically lower than the southern hemisphere by as much as 5 km. The dichotomy is thought to be a giant impact basin, rather than an oceanic plate, and its surface below sedimentary cover is thought to date to the Pre-Noachian [Golombek and Phillips, 2010]. On the edge of the dichotomy, a volcano-tectonic province, the Tharsis rise, covers 15% of the Martian surface and is 8 km high above the surrounding highland terrain. [Andrews-Hanna et al., 2008]

One of the most enigmatic features on the Tharsis rise is a system of deep chasms, Valles Marineris. Valles Marineris has a complex history of formation. Many chasms in Valles Marineris contain enigmatic so-called Interior Layered Deposits [Fueten et al., 2008]. The walls of the chasmata vary widely in their morphology, and include textures as diverse as landslide scars, talus slopes and spur-and-gully [Peulvast et al., 2001]. Whether spur-and-gully texture is a purely erosional feature or if it is tectonically controlled, remains a subject of debate [e.g., Patton, 1990].

Study Objectives

The purpose of the study is to apply the new set of tools to two different data sets of differing resolution to investigate if the tools can aid in the geological interpretation of DTM data. We will investigate:

1. Use of 2nd derivative for a replication of the Knapmeyer et al. [2006] mapping of compressive and extensional features throughout the Martian crust.
2. Use of 1st derivative for the analysis of spur-and-gully slope orientations in Valles Marineris.
3. Use of 2nd derivative for the comparison of curvature orientations within the chasmata of in Valles Marineris and a region of extension faulting in Iceland.

CASE STUDIES

1. Replication of Knapmeyer et al. [2006] effort for mapping compressional and extensional tectonic features across Tharsis rise

To better understand the spatial distribution of Mars tectonism, Knapmeyer et al. [2006] compiled an inventory of 8500 faults classified as compressive or tensional tectonic features. The source dataset was a MOLA DTM, resampled at 1 km/px. It was hillshaded at two orthogonal directions and projected as Mercator for equatorial regions and as stereographic for polar regions above 60° latitudes.

The features were mapped manually using GRASS GIS from hillshaded images, with grabens and half grabens mapped as tectonic extension, and wrinkle ridges and lobate scarps mapped as compression (Fig. 51).

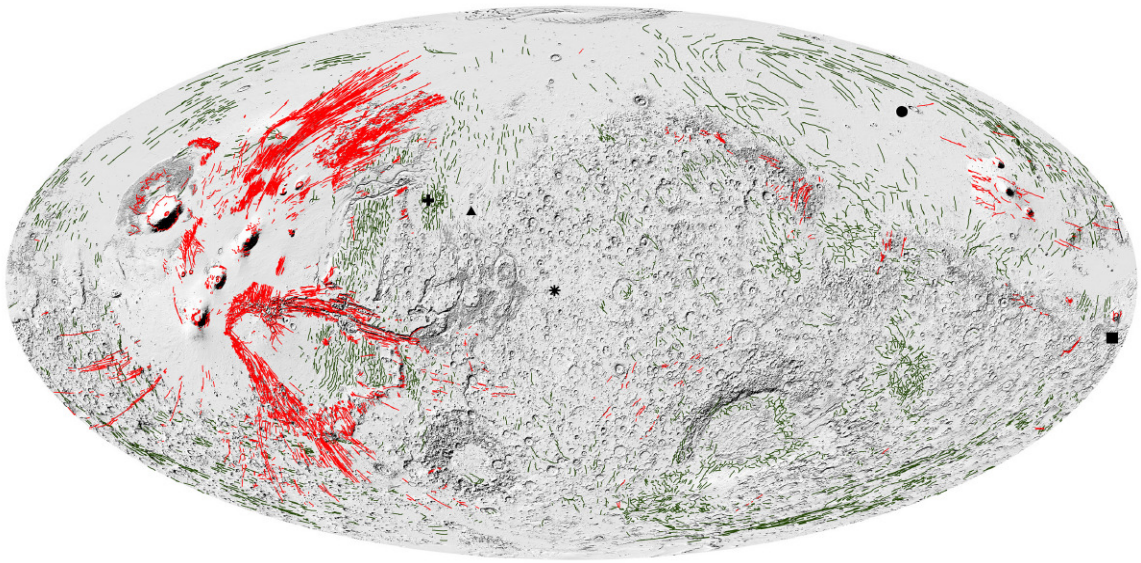


Figure 51: Global distribution of compressive and extensional features, with extensional features shown in red and compressional in green [after Knapmeyer et al., 2006]

Here, we attempted to replicate their manual effort automatically, using the same MOLA DEM and the second derivative tool developed here. The input datasets used were a MOLA DEM at 463 m/px with vertical resolution of 1m. The software used for making of the second derivative was the Python Script Convolution tool developed here and the relevant ArcTools models. The assumption we made was that compressive features can be identified as primary bulges (i.e. lineations of negative curvature), and extensional features as primary troughs (i.e. lineations of positive curvature). On the DEM these appear as lines of higher/lower than average elevation, on AVA these features are distinguishable from changes in colour. (Fig. 52)

While the manual identification of a ridge is relatively easy, the curvature computation yields a more complex result. On a curvature raster, troughs appear as a positive curvature line surrounded by a negative curvature lines on either side. Similarly,

a ridge is represented by a line of negative curvature with two lines of positive curvature on either side (Fig 54, left).

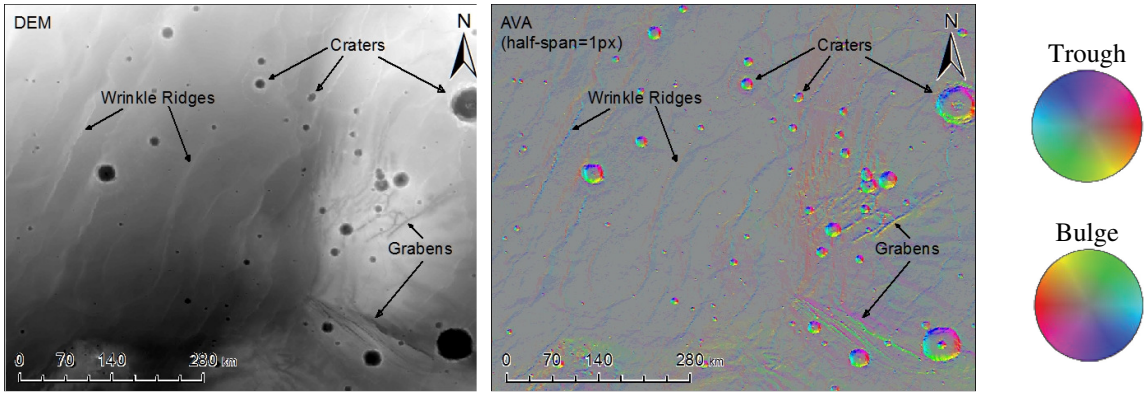


Figure 52: Left: a sample segment from MOLA DEM from Thaumasia Planum, showing wrinkle ridges, craters and grabens. Middle: the first derivative AVA from the same region with a square kernel at a half-span of 1 px, i.e. 3x3 kernel. Right: direction colour key for AVA of concave up (troughs) and concave down (bulges) half-spheres.

Hence, distinguishing extensional and compressional features with a human eye is easy; however, doing so automatically becomes problematic (Fig 53).

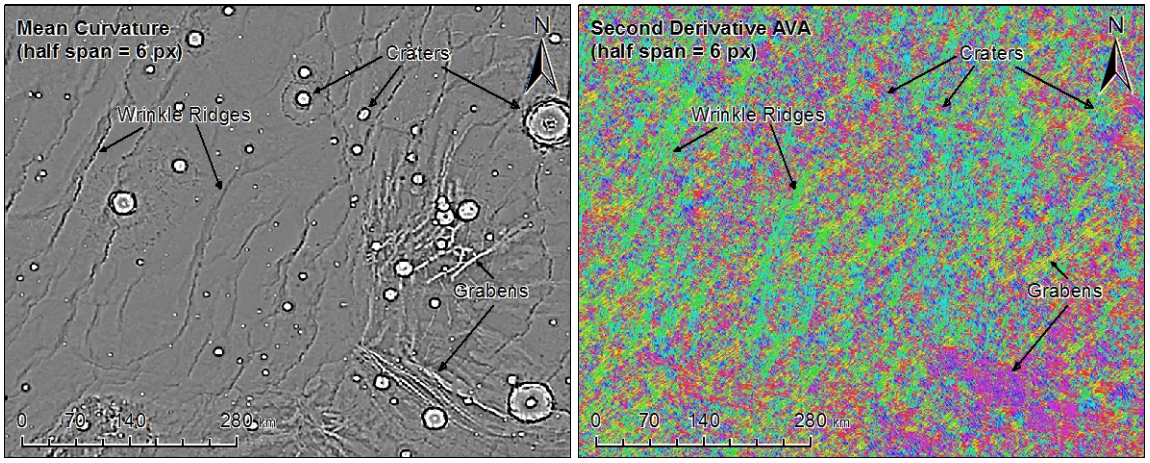


Figure 53: Left: Mean curvature. Right: second derivative AVA. Both images produced with a circular kernel of 6 px radius

The derivatives of elevation are such that the curvature peaks at the elevation extrema (minimum or maximum) and is zero at the elevation inflection. The slope, on other hand, peaks at the elevation inflection and is zero at the elevation extrema (Fig 54). The slope is calculated as the magnitude of a vector sum of gradients, calculated as $G = \sqrt{(G_x^2 + G_y^2)}$.

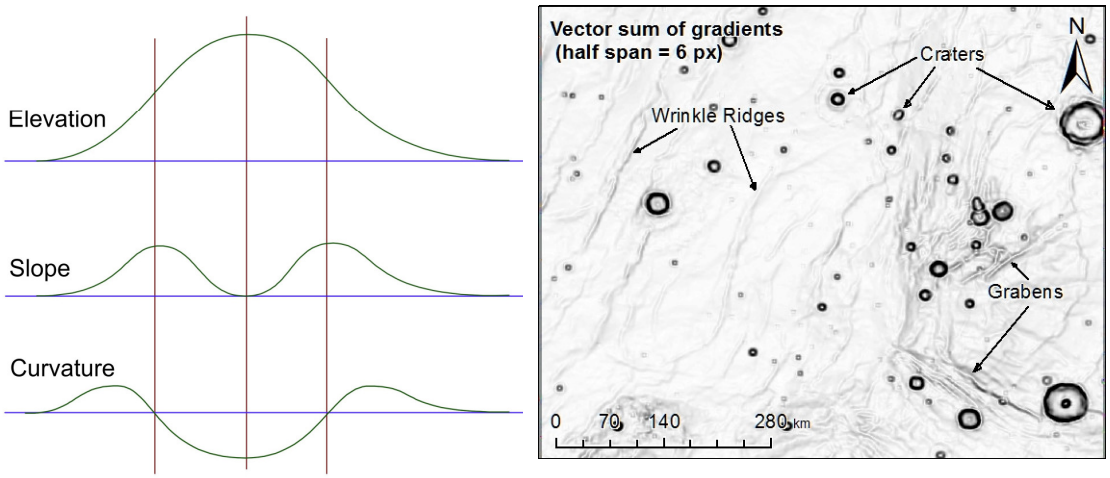


Figure 54: Left: Schematic comparison of elevation and its derivatives of slope magnitude and mean curvature;
Right: Vector sum of gradients (i.e., slope magnitude) raster.

Consequently, it is possible to remove curvature signatures from zones corresponding to slope inflection simply by dividing the mean curvature (Fig 55) by the vector sum of gradients. Dividing slope by curvature would emphasize the areas of high curvature and low slope. (Fig. 56).

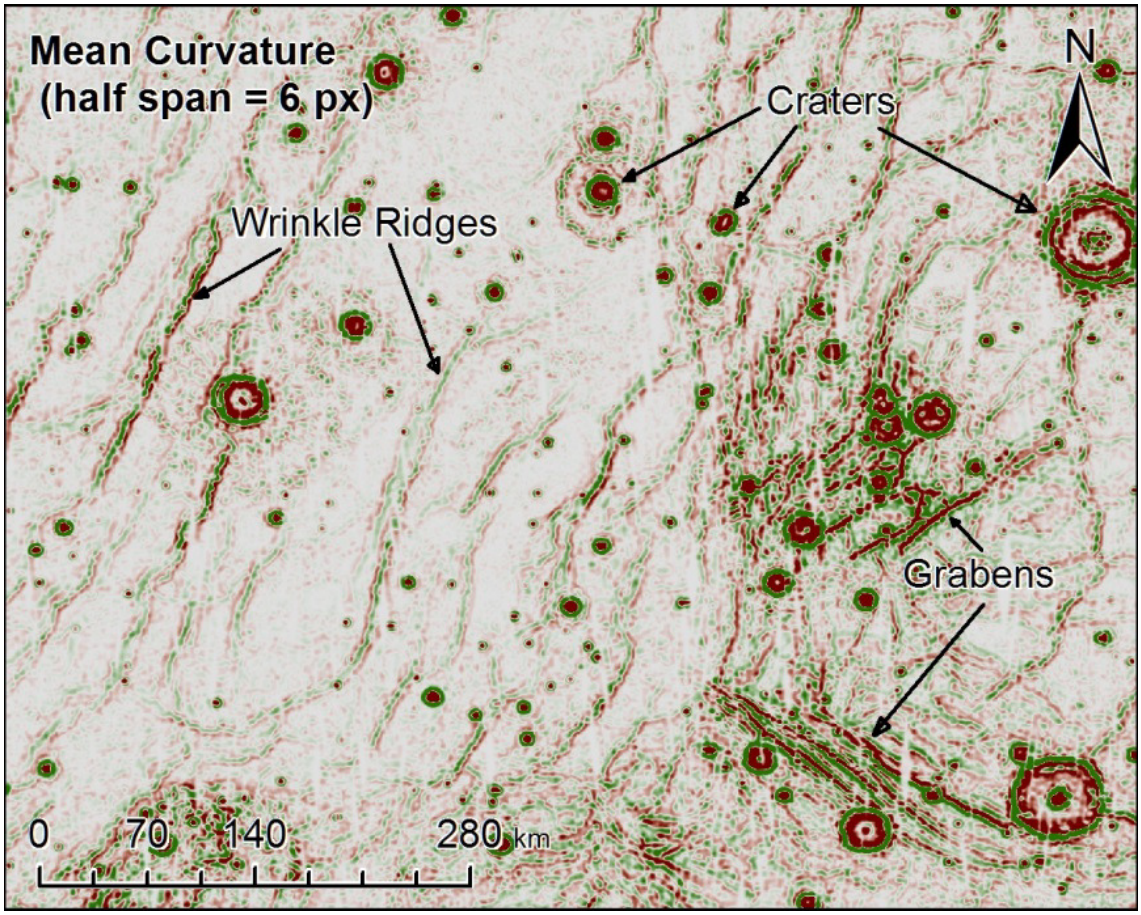


Figure 55: Mean curvature, produced with a kernel of 6 px radius. Negative values (convex, ridges) represented in green, positive values (concave, troughs) represented in red.

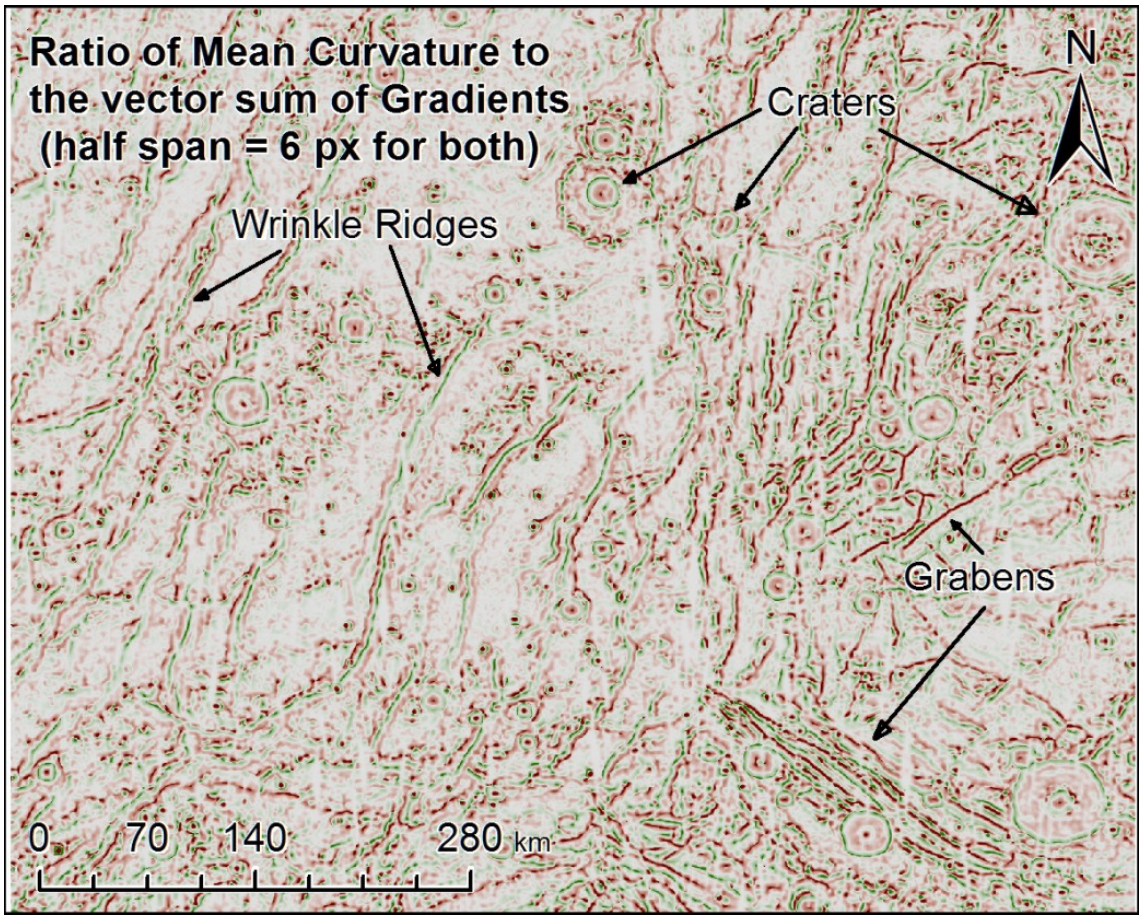


Figure 56: Ratio of mean curvature to the vector sum of gradients, produced with a kernel of 6 px radius. Negative values represented in green, positive values represented in red.

After dividing mean curvature by the vector sum of gradients, low-pass filtering was performed over the output dataset using ArcGIS Spatial Analyst "Focal Statistics". A mean filter with a circular kernel of a 3 px radius was used to reduce noise produced by small localized topographic features. The dataset was converted to binary by thresholding to a manually selected value, such that individual troughs and ridges were represented as isolated features.

The ratio of mean curvature to the vector sum of gradients data is very noisy (small isolated features). To isolate meaningful information, the Spatial Analyst "Thin" (skeletonization) and "Expand" (dilation) tools of the Generalization toolset were used. Repeatedly applying these tools in alteration on a binary dataset allows to elongate linear features far enough that the isolated fragments become reconnected. Randomly oriented segments have lower chance of becoming connected into long chains; hence, subsequent filtering of polyline features by length preferentially selects real features over noise (Fig. 58). This was done via a feedback loop in an ArcGIS ModelBuilder (Fig. 57).

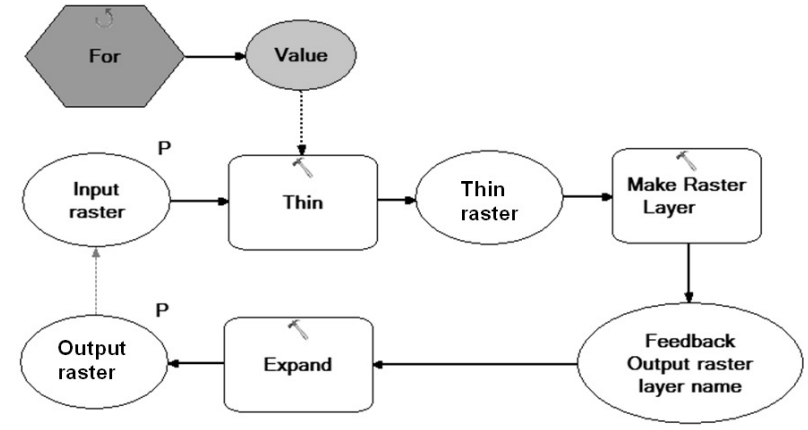


Figure 57: ArcGIS model used to reconstruct fragmented lineations

After five iterations, the raster was then converted to polylines (Fig. 58 a). The dataset was then filtered via selecting by attributes the longest 25% of the features, in order to eliminate noise (Fig. 58 b).

Extension can form both ridges and trough morphologies (e.g., horst and graben topography), but is unlikely to produce ridges alone [Davis and Reynolds, 1996]. Hence, ridges without troughs are most likely the result of compression. To account for this,

lineations representing extensional features (red) were plotted over those representing compressional features (green).

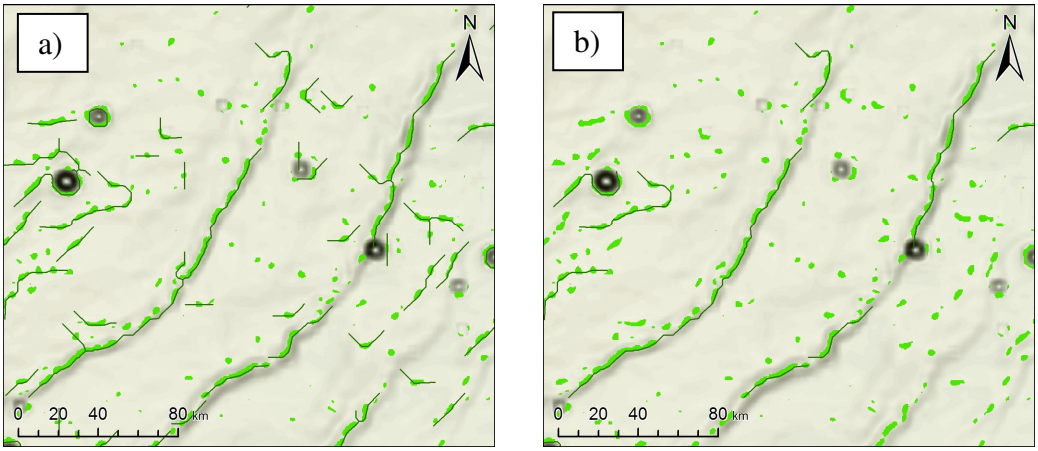


Figure 58: Ridges binary dataset, areas identified as ridges are in bright green, ridges defragmented and vectored in dark green, basemap is the same as figure 44; a) all identified ridges; b) only the longest 25% of the ridges

Results

Extensional and compressional features were mapped in an attempt to replicate the work done by Knapmeyer et al. [2006]. We used the same dataset and the same colour scheme and compared the region of the Tharsis Rise, Mars (Fig. 59). Features plotted were polylines obtained by the process described above. The procedure to produce the polylines still generated a number of artifacts resulting from false alignment of spurious data. Complete removal of these artifacts would require additional processing. It is beyond the scope of this thesis to process such features.

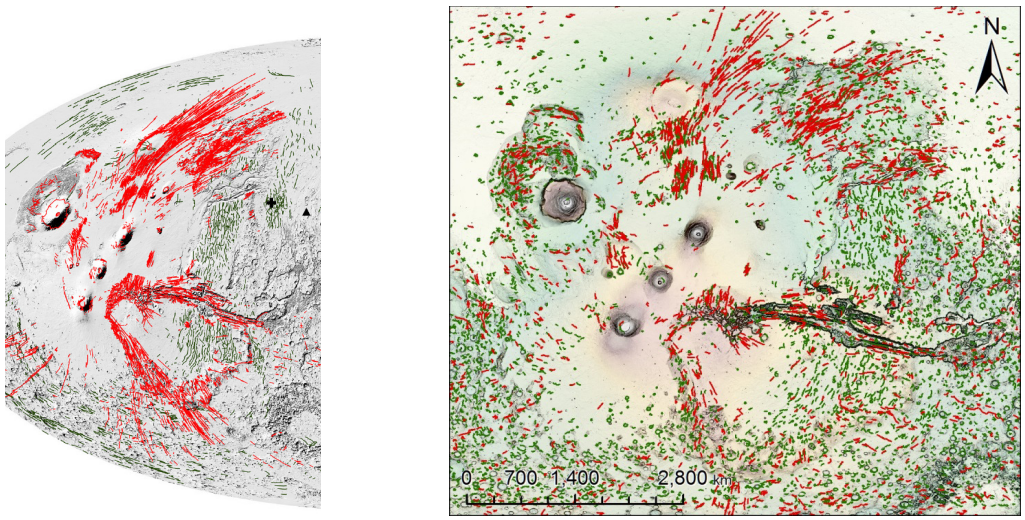


Figure 59: Left: After Knapmeyer et. al [2006], fault expressions were manually identified. Right: replication of the experiment using software described here.

Discussion

The results appear roughly similar to those presented by Knapmeyer et al. [2006], trends generally agree. Troughs tend to be linear, while ridges are generally less continuous and irregular. A possible explanation for this is that ridges, being protruding features, tend to be degraded by erosion, while troughs tend to get filled. Hence, to obtain optimal results, thresholding the raster dataset representing the ratio of mean curvature to the vector sum of gradients, must be done using different cut-off values for ridges and troughs. No attempt is made in this work to find an analytical solution to optimal thresholds; they were determined by trial and error. The effectiveness of this method appears to depend strongly on the roughness of the terrain, as cratered areas produce significant noise. This method has a good potential, but still requires manual expertise.

This procedure is certainly not the only solution to automatically distinguish extensional and compressional features from a DEM using the tools presented in this

thesis, and is probably not even the best one. Other approaches were considered but not evaluated due to time constraints, include:

- Using Mean Signed Squared Differences (MSSD) derivative, might be useful because this goodness-of-fit estimate shows whether there are points in a neighbourhood far above or below the plane of best fit.
- Taking a gradient of a strike raster, might be useful because strike direction changes by 180° when passing over a ridge or a trough.

Furthermore, assuming that negative curvature at elevation local extrema always represents compression and positive curvature always extension, is only a very crude approximation. Compressional stresses may produce locally concave up topographic features (i.e. troughs), in case of backwards buckling, subsidence on a footwall, or erosional destruction along the fault plane. Similarly, extensional stresses will often produce horsts in addition to grabens, which would appear as topographically concave down (i.e. ridges). A more complex identification procedure, accounting for these issues would produce more accurate and extensive results.

2. Spur-and-Gully slope orientations

While many of the chasmata have wall sections that display a strong spur-and-gully texture [Peulvast et al., 2001], Hebes chasma was chosen for this study for the sake of consistency with the previous chapters (for location see Fig. 50).

The input dataset used was 50 m/px HRSC mosaic DEM covering Hebes Chasma with vertical resolution of 1 m. The first derivative was obtained with a 6-px half-span (11x11 px) square window using All-In-One tool developed earlier. The quantitative data was analyzed using SpheriStat (© Pangaea Scientific).

Attitudes were obtained using a HRSC 50 m/px DEM of Hebes Chasma. The walls of the entire chasma were cropped via “Extraction by Mask”, and then a small section of the walls with a well developed spur-and-gully expression, was selected for further study.

Appearance of the AVA seems to indicate that some of the surfaces may be planes of the same attitude; indeed, both north and south walls display some blue and some orange colouring; and it even seems that some of the red-coloured planes could even be traced across the gullies (Fig. 60).

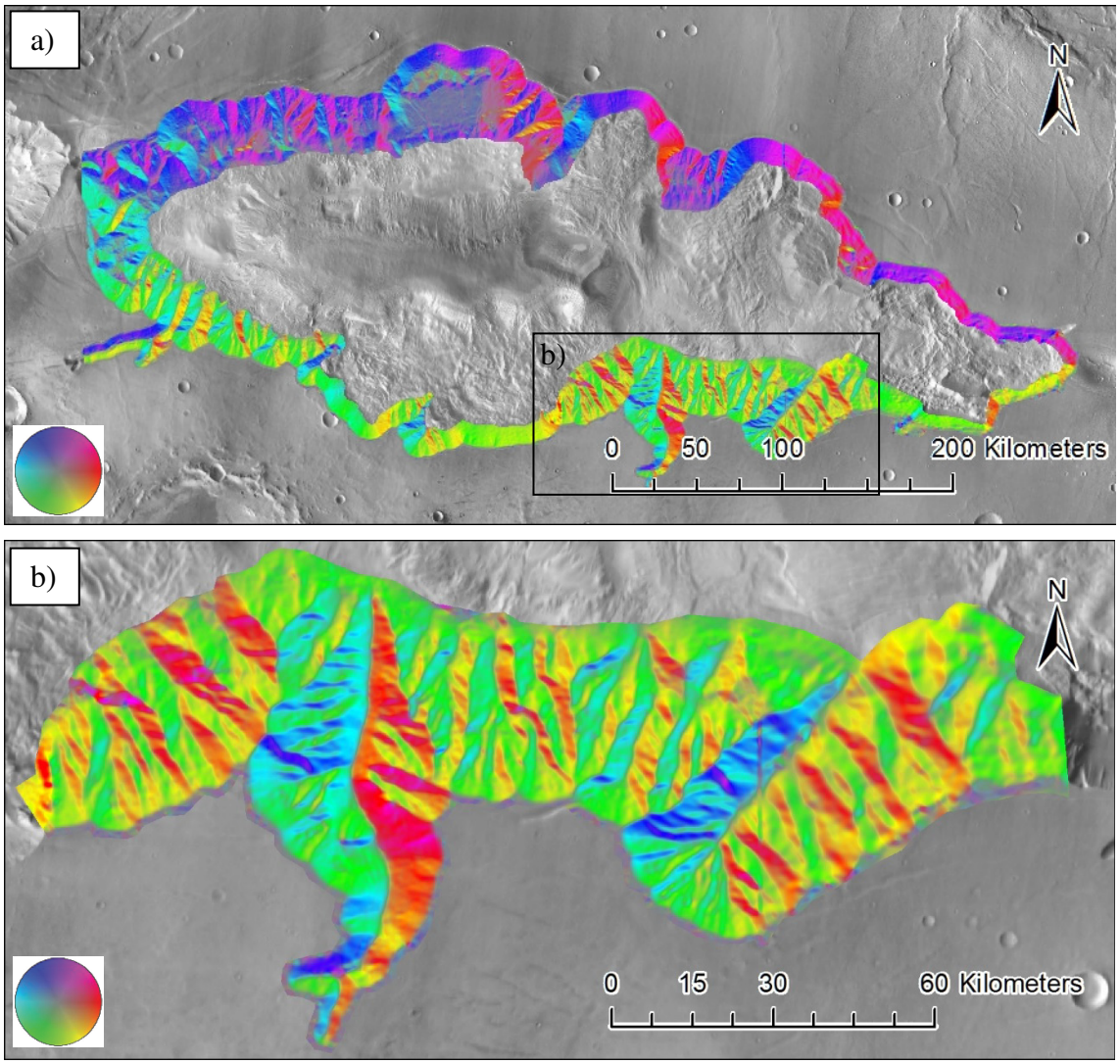


Figure 60: First derivative AVA showing regions whose attitude quantitative data was analysed with SpheriStat (© Pangaea Scientific). AVA of a concave-up half-sphere as colour map in the bottom left corner.
a) Walls of Hebes Chasma;
b) A section of that wall on the south side displaying spur-and-gully morphology.

To determine whether the attitude data shows a pronounced frequency distribution peaks, pixel counts for each occurrence of particular strike/dip value were calculated and plotted on a stereonet using SpheriStat (© Pangaea Scientific).

This analysis was achieved by using the raster calculator, Spatial Analyst, extension, where a 16-bit unsigned integer dataset was produced following the formula *"Value = Int(Strike*100+Dip)"*. The attribute table of the dataset contains the frequencies of occurrence (counts). Counts represent a total number of times a value has occurred, i.e. an attitude that occurs in N pixels on a raster would have a count of N in its attribute table. The counts are necessary to reduce the total number of data points that will be supplied to SpheriStat (© Pangaea Scientific). Since ArcGIS follows a different format when exporting numbers below above 1000 (numbers above 1000 are surrounded with double quotation marks and use coma for a thousands separator), a weighting factor was calculated by dividing instance counts by 10 to keep all numbers below 1000 and avoid errors when importing to SpheriStat (© Pangaea Scientific).

Results

New fields were added to the attribute table and populated using Field Calculator, following the formulas *"Strike = Int(Value/100)"* and *"Dip = Int(Value Mod 100)"*. The unrequired fields of the attribute table were turned off, and the selected data was exported as a text file and subsequently imported to SpheriStat (© Pangaea Scientific) to compute the weighted point density plots (Fig. 61).

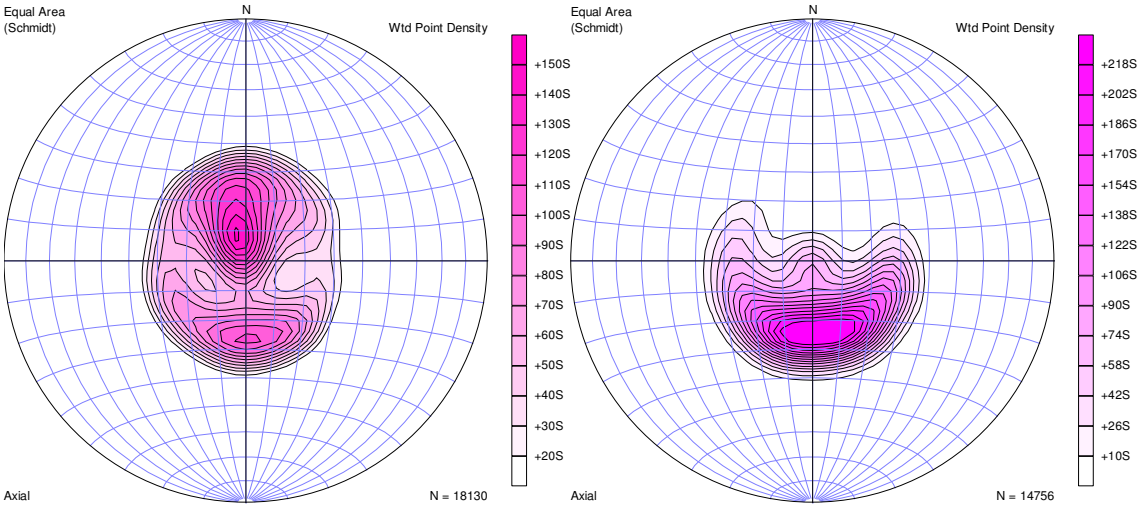


Figure 61: Weighted density plots produced with Spheristat (©Pangea Scientifc) using *Gaussian - K=100* counting model. Left: for the walls of Hebes chasma; right: for a Southwest section of the walls which displays a distinct spur-and-gully texture (for location see fig. 60).

The frequency distribution for the chasma wall appears bimodal on the *Gaussian K=100* stereonet frequency plot. For the entire Chasm there are two statistically significant peaks corresponding to 263°/25° and 071°/10°. For the segment of spur-and-gully texture to the Southwest there is one statistically significant peak at 263°/25°.

To identify possible individual smaller sub peaks, the Gaussian E=3xSigma counting method was used, and contour levels were adjusted to produce several distinct peaks (Fig. 62). The attitudes of these peaks are consistent between the two stereonets. These peaks are not statistically significant; however, they may still be indicative of preferential orientations.

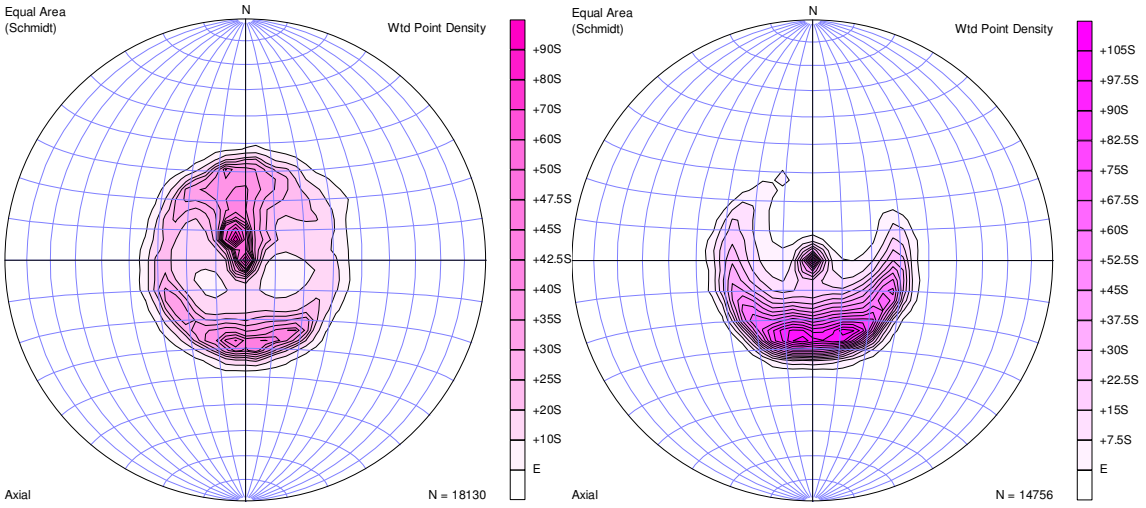


Figure 62: Weighted density plots using "Gaussian – E = 3 x Sigma" counting model with extra fine grid density and contouring intervals shown to the right. Left: Orientations weighted by frequency, processed for the entire Hebes Chasma walls, regardless of the wall type (region shown in Fig. 60 top); right: for a southern section of the Hebes Chasma wall with spur-and-gully topography (shown in Fig. 60 bottom).

Discussion

Stereonet density distribution contour plots (Fig. 62) show several steep peaks, implying that a high proportion of the ridge sides share common orientations (i.e., they are more-or-less planar). Erosional ridge would exhibit a broader peak in attitude.

Six peaks have been distinguished from contour plots. The peak at 0° dip corresponds to flat areas of the plateau. The broad peak at 036°-090° / 06°-15° corresponds to what has been interpreted as a Landslide Scar [Schmidt, 2015] on the north wall centre. The landslide Scar terminates abruptly at the basal scarp, the planar morphology of this erosional feature is likely caused by tectonic controls exhibited by the basal fault. The remaining four peaks all are relatively steep, with dip values over 25°, their strikes are 075-082, 236-249, 272-285, and 314-333. These seem to represent the preferred directions of planes on spur-and-gully texture walls (Fig. 64).

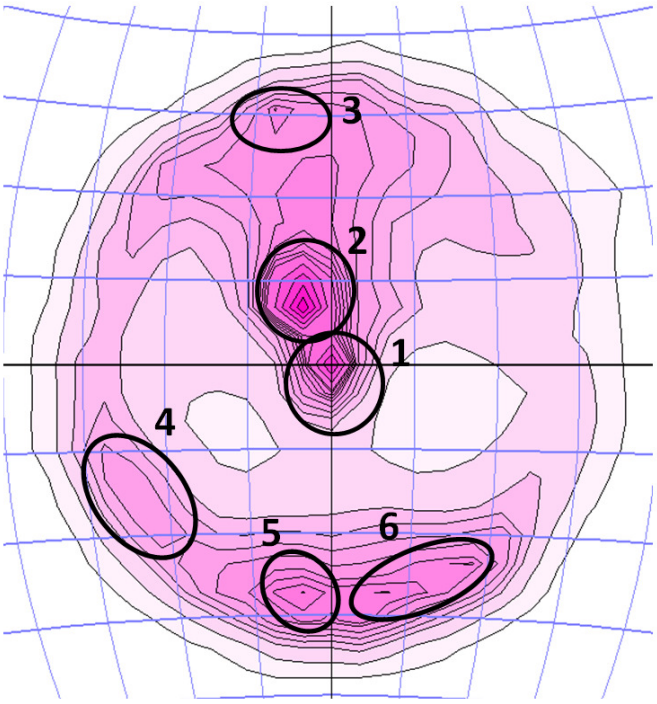


Figure 63: A zoomed-in view of the weighted density plot using "Gaussian – E = 3 x Sigma" counting model for the walls of the entire Hebes chasma, with individual sub peaks manually identified and ladled.

The dips on the surfaces exposed for a long time, which is the case for the chasmata walls, are restricted to be below the angle of repose determined by erosional processes [Mege and Masson, 1996]. Angle of repose is independent of the acceleration due to gravity, and is thus the same for Mars as it is for Earth, approximately 33° [Atwood-Stone and McEwen, 2013]. Consequently, if the steep dipping surfaces are indeed tectonically controlled, the underlying faults could be much steeper than their erosional expressions.

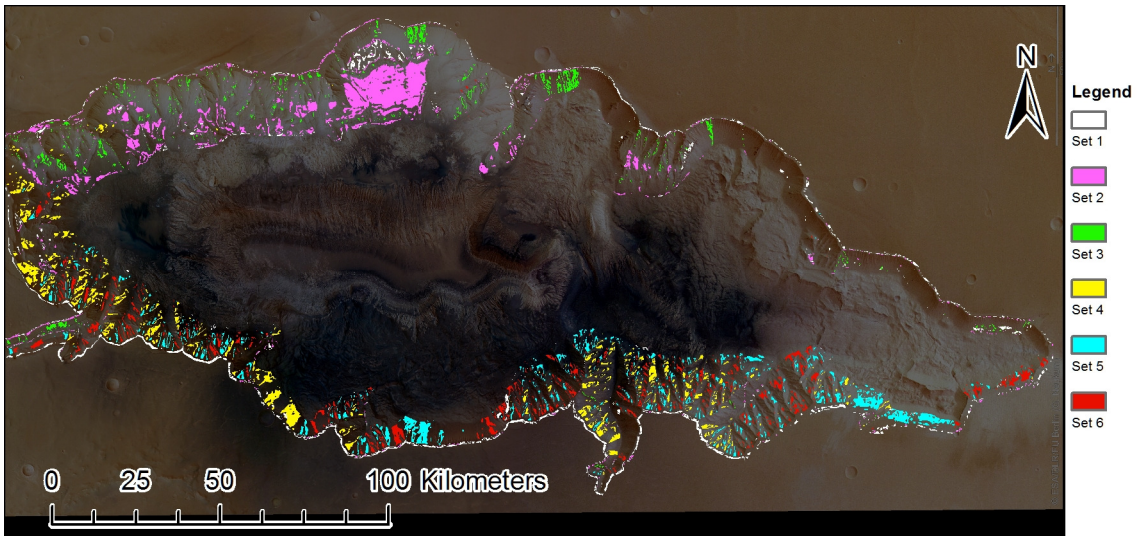


Figure 64: Sets of surfaces corresponding to peaks observable on stereonet density distribution contour plots shown in figure 63. Set 1: Dip 0°-4°; Set 2: strike 036°-090°, dip 06°-15°; Set 3: strike 075°-082°, dip >25°; Set 4: strike 314°-282°, dip >25°; Set 5: strike 272°-285°, dip >25°; Set 6: strike 236°-249°, dip >25°.

The surface orientation counts are biased towards north- and south-dipping directions due to elongated shape of the chasma. Erosional processes would imprint preferences for north-dipping surfaces on the south wall and for south-dipping surfaces on the north wall, and Hebes chasma does not have well developed east and west walls.

A purely erosional mode of formation would have produced a broad bimodal distribution. However, presence of several high and narrow peaks (Fig. 63), which, when displayed on a map (Fig. 64), appear to correspond to facets of Spur-and-Gully wall textures, suggest a presence of underlying structural controls. Hence, the most likely control of the spur-and-gully morphology is underlying faults.

3. Comparison of tectonic environments based upon curvature orientation frequency distribution analysis

A surface subjected to tectonic stress would develop topographic features such as grabens and horsts; or ridges, scarps and folds. On a curvature raster these appear as oriented textures. Trends of these features can be seen on a curvature orientation derivative of the DEM. Looking at the shape of the rose diagram, and, in particular, the angle between the prevailing curvature directions, may help distinguish between different tectonic regimes, for instance strike-slip faulting would trend approximately 60° from a wrinkle ridge, and 30° from a graben [Andrews-Hanna et al., 2008].

In this study, curvature orientations of the canyon walls of Valles Marineris were investigated. These were compared against Iceland topography, which was chosen because of the relatively simple tectonic history of Iceland. The input datasets used were a 50 m/px HRSC mosaic DEM covering Valles Marineris and vertical resolution of 1 m/px. The Iceland dataset consists of an airborne HRSC DEM of Iceland, obtained from E. Hauber (DLR, location unknown) [personal communication, 2015] with horizontal and vertical resolutions of 1 m/px and 0.1 m respectively. The software used in production of second derivative was the Python Script Convolution tool developed here and relevant ArcTools models. Settings for second derivative were with a 51-px radius circular kernel, because the larger kernels are less susceptible to noise.

The data for each individual chasma was extracted using manually drawn polygon mask (Fig. 65).

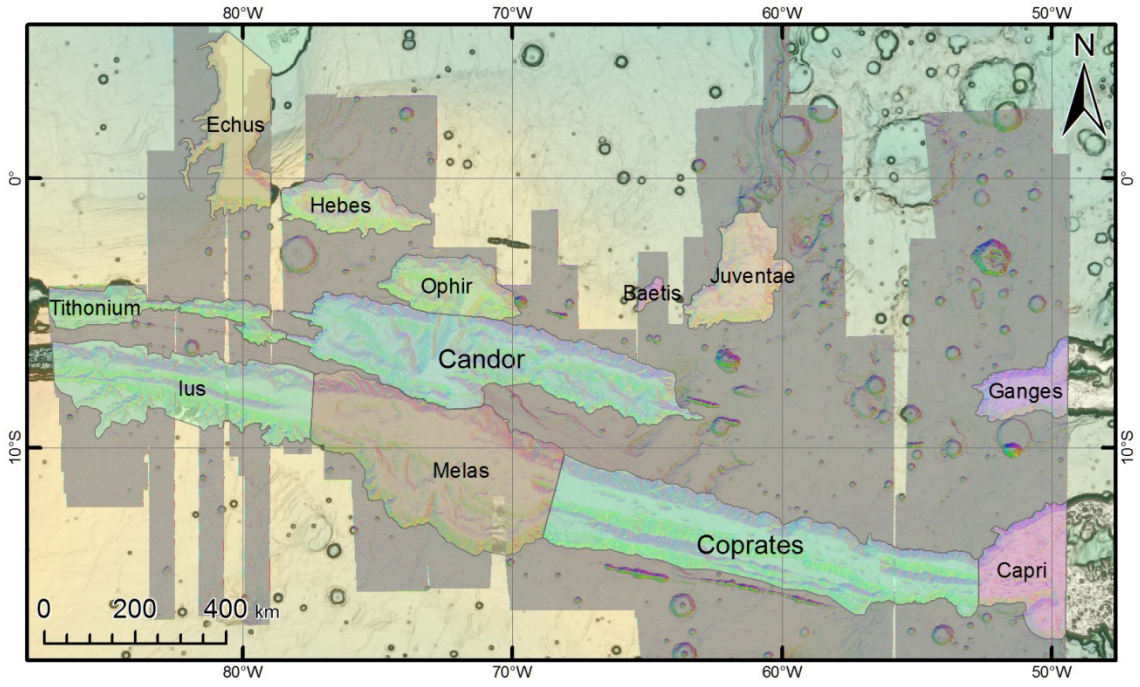


Figure 65: Polygons used to crop datasets of Valles Marineris chasms and HRSC DEM coverage overlaying the base map produced in the first subsection.

Cumulative counts were obtained from the attribute table; data was duplicated to produce the second half of the rose diagram (180°-360°) and normalized by dividing through by the total number of pixels in each dataset. Thus, rose diagrams for curvature directions were built in Excel to show frequency distributions in terms of the percentage from the total. Rose diagrams were plotted on a 360° wheel, by means of Excel Radar chart type.

Iceland is a region currently undergoing tectonic extension (Fig. 66) and this data was used to compare with the Martian chasm data.

Results

Results for this study are shown in Figure 66 and 67. The Iceland data (Fig. 66) shows very smooth, symmetrical, sharply peaking unimodal distribution.

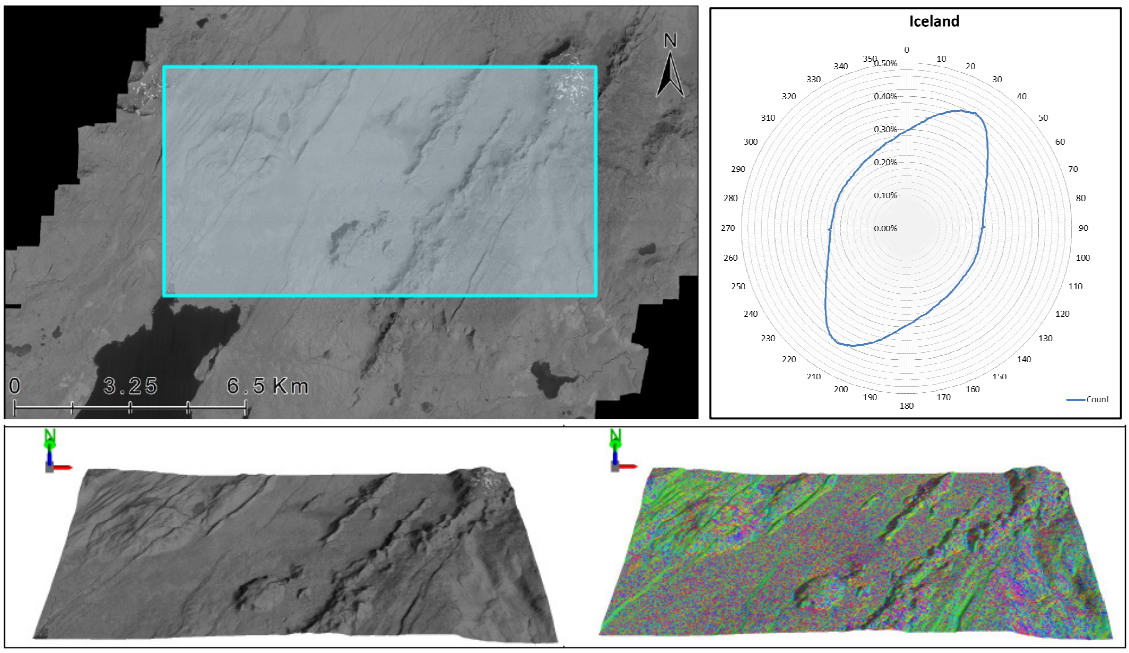


Figure 66: Curvature orientations for terrestrial example, Iceland, obtained using a kernel with 51 px radius. Top left: input raster DEM, location unknown, blue box shows the sampled region. Bottom left: 3D projection of the sampled region DEM. Top right: rose diagram for frequency distribution of orientation, axial data plotted in Excel. Bottom right: 3D projection of the sampled region second derivative AVA.

By contrast, the frequency distributions of Chasmata in Valles Marineris show considerable variation and more complex shapes (Fig. 67).

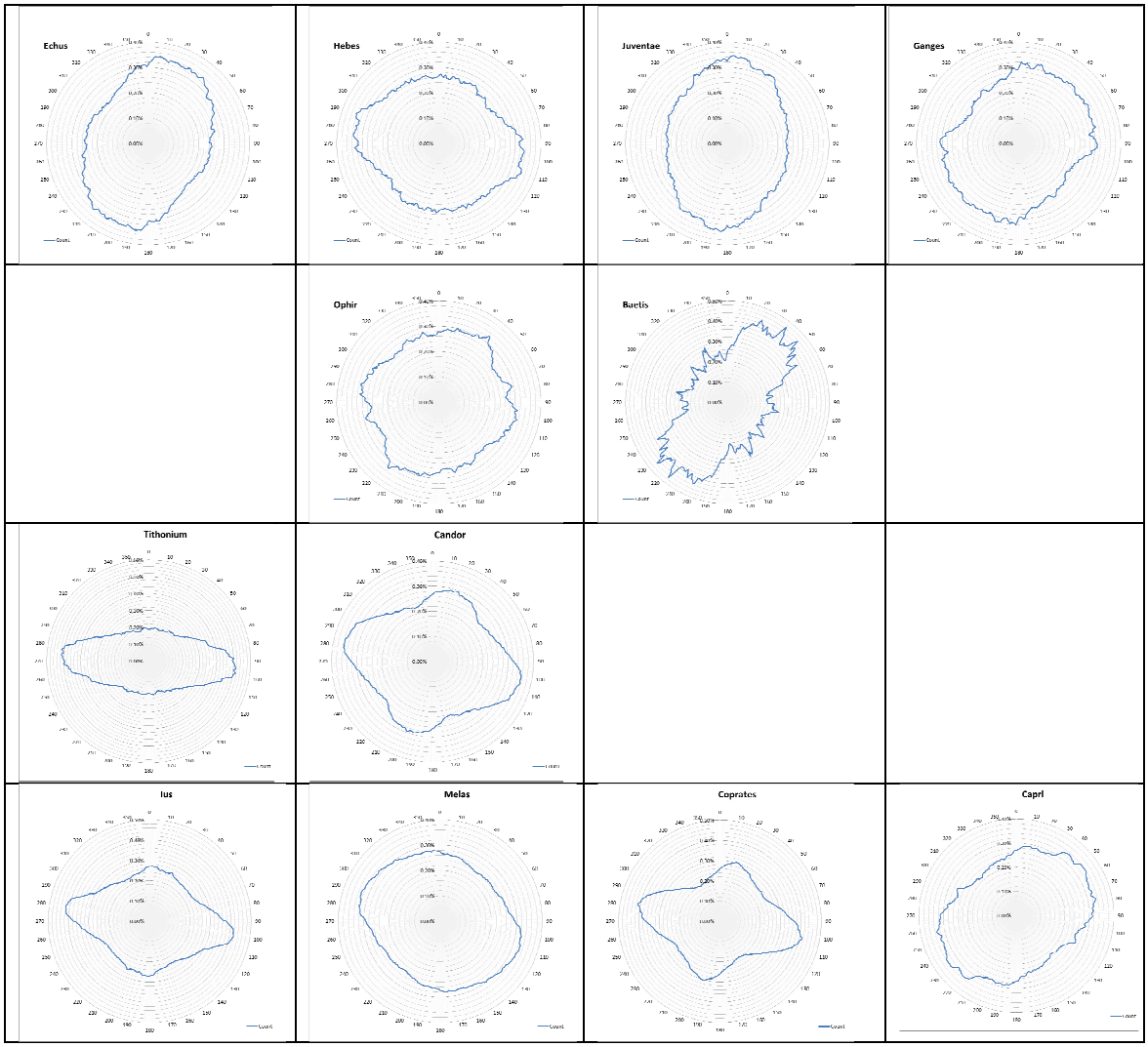


Figure 67: Frequency distribution of curvature orientations for the entire chasm for each of the chasmata; diagrams span 360°, layout is to mimic the geographical arrangement of the chasmata (see Fig. 65 for map of Valles Marineris).

Discussion

The tectonic environment in the Iceland sample region is known to be a simple extensional system, topographically expressed as a horst and graben morphometric

texture. The diagram has a slight peak at 90°, which parallels the flight path, and thus interpreted as an artifact of systematic error that originated at data collection stage.

Of all the chasmata (Fig. 67), only Tithonium shows a strong unimodal distribution similar to that which is seen in Iceland example. It may be the case that two have very similar tectonic regimes, which would imply that Tithonium is predominantly a graben formed in a simple extensional setting; however, further study is needed to establish whether this is the case.

Distributions of Echus and Juventae are nearly unimodal; however, they are slightly skewed and have broader peaks than the Iceland example. This may indicate a more complex geological history, such as interplay of several subparallel fault directions. The principal direction for Juventae is 010° while it is 025° for Echus. The Echus peak is very broad, with the frequency staying nearly constant from 005° to 040°. This may indicate overlapping of several peaks.

Ganges and Capri distributions are very similar. Both appear unimodal with very broad peaks. The broadness of frequency distribution peaks may indicate prevalence of erosional controls over tectonic. Capri chasma has a small, yet sharp peak at 115° azimuth, the significance of which is unknown.

Principal directions on Melas, Hebes, and Ophir are not strongly pronounced. Melas appear unimodal, but strongly skewed, while Hebes and Ophir are weakly bimodal. Ophir contains several sharp yet minor secondary peaks, which, if statistically significant, may indicate additional a more complex environment.

Unusual peaks similar to the ones seen in Ophir also appear in Baetis chasma, and are more pronounced there. Baetis shows three secondary peaks, which may imply either a more complex tectonic history or lesser degradation of tectonic signatures by erosion.

Ius, Candor and Coprates appear to have frequency distributions of nearly identical shape and orientations. They show bimodal distributions almost at 90° from each other. These chasmata were further analyzed by selecting only their north walls (Fig. 68)

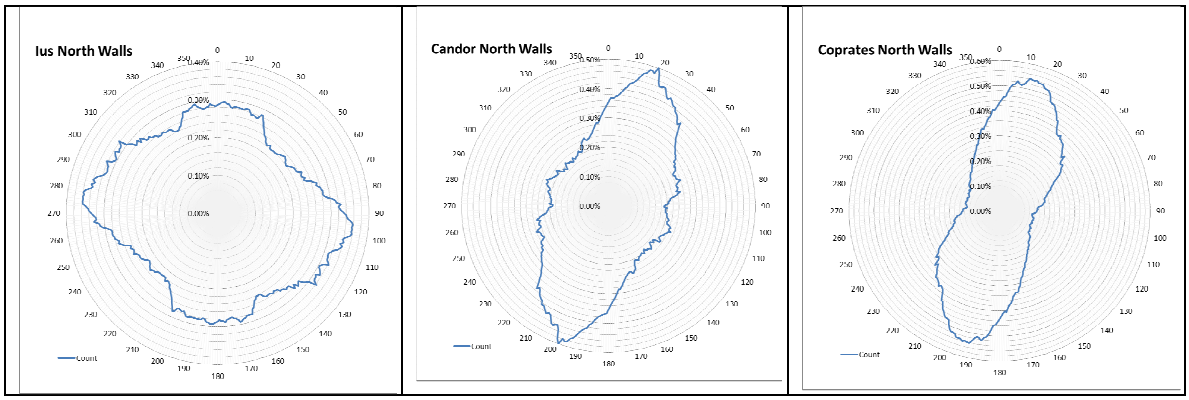


Figure 68: Curvature orientation on north walls on Ius, Candor, and Coprates chasmata. The diagrams span 360°

All three appear to have a peak at 010°-020°. Coprates north wall shows strongly unimodal distribution, Candor north wall appears trimodal, with two minor peaks in addition to the principal - a second peak at 075° and a third one at 110°. Ius north walls show a strongly bimodal distribution with broad peaks at 005° and at 110°. This does not necessarily signify change of the tectonic environment, since the peak which disappears from West to East is the general direction of Valles Marineris.

CONCLUSIONS

The method shows promise and more work is needed to explore its full potential. Curvature and Slope derivatives allow for generation of quantitative data pertaining to topographic feature orientations. This data can be exported and analyzed with third party software. The raster datasets of DEM derivatives can be used for feature registration and

segmentation, such as generation of polygons covering planar surfaces, or polylines representing ridges and troughs.

There are limitations to this methodology, such as long processing times associated with the use of large kernels, and high susceptibility of derivative quality to noise in the input DEM. Feature registration may be completely impeded by fragmentation due to noise or overlap of multiple features. Attempting to defragment the features, as was described in subsection 1 of this chapter, invites a risk of registering a false positive for a feature that is not there due to false alignment of spurious data.

The tools and methods presented in this thesis are useful and effective, yet still require human expertise to ensure quality results.

Chapter 6: Conclusions

Analysis of topography can provide insight onto the processes that formed it. The most commonly used Geographic Information System (GIS) is ESRI ArcGIS. ArcGIS contains Spatial Analyst extension designed specifically for analysis of topography (i.e., geomorphometry); however, ArcGIS has some limitations. This thesis presents useful routines for analysis of topographic and 3D surface imagery. New code to run within ArcGIS was developed and compared with routines publically available. Beneficial techniques were developed and their application to a number of different problems demonstrated. This work produced geoprocessing tools for conducting morphometric analysis on Digital Elevation Models. The software developed was designed specifically tailored for structural geology applications to be accessible from ArcMap environment.

We developed two programmatic solutions to the computation of surface first derivative. An ArcGIS command tool and an ArcGIS toolbox. The command tool is discussed in chapter 2, while the toolbox is elaborated on in chapter 3. Chapter 4 elaborates on the computation of second derivative, and higher order derivatives. Chapter 5 demonstrates applications of these tools to real geological problems.

The techniques for the first derivative include computation of surface gradients, trigonometric conversion of gradient data into strike and dip representation, and various approaches to estimate the goodness-of-fit of a derived plane. The goodness-of-fit estimates provide a nice way to estimate surface roughness, and can also be effectively used to identify errors in the dataset.

A technique of representing both strike and dip in one visualization as hue and saturation has been implemented. It is noted that such representation does not provide

most optimum viewing experience due to perceptual bias caused by differences in relative luminance of different colours displayed on different devices. A capability for applying user-defined colour maps for visualization is provided.

Techniques described in Chapters 2 and 3 produce identical results regarding the orientation of the surface. The tools in chapter 3 are much faster, but the tool from chapter 2 also allows differentiation of parallel surfaces from coplanar. Tools in chapter 3 implement convolution filtering, and provide means for convolution filtering, which is currently unavailable in ArcGIS. Implementing these procedures as an ArcGIS toolbox eliminates the need for installation. This means the they can be used out-of-the box without administrative privileges. These tools are scripts, and hence can be readily modified by the end user.

The chapter 4 investigates use of convolution filtering for computation of surface second derivatives, which are curvature. ArcGIS provides a tool for computation of curvature, which outputs Profile, Planiform, and Mean curvatures. This thesis on other hand investigates a different form of curvature representation, namely principal curvatures and the degree of declination of principal curvature axis from geodesic. It does so by fitting an equation of a paraboloid.

Unlike the first derivative that is largely unaffected by the shape of the sampling window, the curvature demands the use of circular kernels. The methods of calculating normalization weights for implementation of circular kernel geometries have been discussed in this thesis.

In areas where geologic features such as faults or bedding planes control the topography they can be recognized by surface orientations preference. The topographic expressions of tectonic features can be identified by statistical analysis, or visual

assessment. AVA provides a means for visual assessment, while the Strike and Dip raster data can be easily converted into ASCII representation for quantitative analysis with third party software. In this thesis, statistical analysis of orientation data was conducted using Spheristat (©Pangea Scientific) software. The datasets can also be processed further with other existing ArcGIS tools, providing means for raster segmentation and feature registration (i.e., computer vision – raster to vector data conversion). The tools in the toolbox can be included in other models allowing for extensive automation of the workflows, reducing the demands on software operators.

An important outcome of being able to determine the azimuth of minor axis of principal curvatures is that it describes topographic fabric orientation, i.e., which way the ridges and troughs in the surface are trending. The orientation of ridges and troughs may be indicative of dominant stress orientation in the subsurface. The relative strength of the morphologic fabric can be obtained by comparing the ratio of principal curvature magnitudes. The curvature direction and the principal curvature magnitudes ratio can be used to as inputs into the AVA tool to produce visualization of the second derivative.

The applications of the methods were effectively employed for three case studies presented in Chapter 5. The first case study conducted an investigation into the distribution and orientation of compressive and extensional features in the Tharsis region, Mars. The results were found to correlate well with previous studies conducted by manual examination of shaded relief. The second case study conducted analysis of spur-and-gully slope orientations in the Hebes Chasma, Valles Marineris, Mars. This study discovered several frequency peaks, which may correspond to orientations for potential fault directions. The third study conducted curvature orientation frequency analysis on large number of chasmata in Valles Marineris, Mars, and compared them to the

terrestrial example of tectonic rifting in Iceland. This study concluded that orientation frequency distributions in chasmata of Valles Marineris is far more complex than those produced by simple rifting on Earth, likely indicating a far more complex formation history.

The tools developed in this body of work show great promise and further study will be required to determine their full potential. Some suggestions for the follow-up research would include: improving techniques for raster to vector data conversion and frequency analysis.

References

Atwood-Stone, C., & McEwen, A.S., 2013. Measuring dynamic angle of repose in low gravity environments using martian sand dunes. 44th Lunar and Planetary Science Conference, held March 18-22, 2013 in The Woodlands, Texas. LPI Contribution No. 1719, p.1727

Agoston, M.K., 2005. Computer graphics and geometric modeling. Heidelberg: Springer.

Andrews-Hanna, J. C., M. T. Zuber, and S. A. Hauck II (2008), Strike-slip faults on Mars: Observations and implications for global tectonics and geodynamics, J. Geophys. Res., 113, E08002, doi:10.1029/2007JE002980

Anton, H., & Rorres, C., 2010. Elementary linear algebra: applications version. Tenth Edition. Wiley, USA. ISBN 978-0-470-43205-1

Barraud, J., 2006. The use of watershed segmentation and GIS software for textural analysis of thin sections. Journal of Volcanology and Geothermal Research 154 (2006) 17-33. doi:10.1016/j.jvolgeores.2005.09.017

Behrendt, R., 2012. Introduction to LiDAR and Forestry, Part 1: A Powerful New 3D Tool for Resource Managers. The Forestry Source, September 2012, pp. 14-15. Society of American Foresters.

Borisenko, A.I., & Tarapov, I.E., 1966. Vector and tensor analysis with applications. Translated from the Russian by Richard A. Silverman. Dover Publications, Inc., New York. ISBN 0-486-63833-2

Boyce, W.E., & DiPrima, R.C., 2009. Elementary differential equations and boundary value problems. Ninth Edition. Wiley. ISBN 978-0-470-38334-6

Brewer, C.A., & Marlow, K.A., 1993. Color representation of aspect and slope simultaneously. Proceedings, Eleventh International Symposium on Computer-Assisted Cartography (Auto-Carto-11), Minneapolis, October/November 1993, pp. 328-337.

Broxton, M.J., & Edwards, L.J., 2008. The Ames Stereo Pipeline: Automated 3D Surface Reconstruction from Orbital Imagery. . 39th Lunar and Planetary Science Conference, 2419.pdf

Davis, J.C., 2002. Statistics and data analysis in geology. Third Edition. John Wiley & Sons, USA. ISBN 0-471-17275-8

Davis, G.H., & Reynolds, S.J., 1996. Structural geology of rocks and regions. Second Edition. John Wiley & Sons, Inc. ISBN 978-0-471-52621-6

Dozier, J., & Strahler, A.H., 1983. Ground investigations in support of remote sensing. Manual of Remote Sensing, Vol. 1, pp. 959-986

Fueten, F., Stesky, R., MacKinnon, P., Hauber, E., Zegers, T., Gwinner, K., Scholten, F., & Neukum, G., 2008. Stratigraphy and structure of interior layered deposits in west Candor Chasma, Mars, from High Resolution Stereo Camera (HRSC) stereo imagery and derived elevations, J. Geophys. Res., 113, E10008, doi:10.1029/2007JE003053.

Galbis, A., & Maestre, M., 2010. Vector analysis versus vector calculus. Spinger. ISBN 978-1-4614-2200-6

Golombek, M.P., & Phillips, R.J., 2010.Mars Tectonics. In Planetary Tectonics, edited by T. R. Watters and R. A. Schultz. Cambridge University Press, UK. ISBN 978-0-521-76573-2.

Gonzalez, R.C., & Woods, R.E., 1992. Digital image processing. Addison Wesley, USA. ISBN 0-201-50803-6

Goodchild, M.F., and K.K. Kemp, eds. 1990. NCGIA Core Curriculum in GIS. National Center for Geographic Information and Analysis, University of California, Santa Barbara CA.

Goodchild, J.S., & Fueten, F., 1998. Edge detection in petrographic images using the rotating polarizer stage. Computers & Geosciences Vol. 24, No. 8, pp. 745-751. doi:10.1016/S0098-3004(98)00054-5

Hargitai, H., & Kereszturi, A., 2015. Spur-and-Gully. Encyclopedia of Planetary Landforms, pp. 2046-2052, Springer-Verlag New York. ISBN 978-1-4614-3134-3

Heywood, I., Cornelius, S., & Carver, S., 2006. An introduction to geographical information systems. Third Edition. Pearson Prentice Hall, Spain. ISBN: 978-0-13-129317-5

Horn, B.K.P., 1981. Hill Shading and the Reflectance Map. Proceedings of the IEE, Vol. 69, No. 1.

Jain, R., Kasturi, R., & Schunk, B.G., 1995. Machine Vision. McGraw-Hill, New York. ISBN 0-07-032018-7

James, G., & James, R.C., 1992. Mathematics dictionary. Fifth Edition. Chapman & Hall, New York, IBSN 0-412-99031-8

Knapmeyer, M., J. Oberst, E. Hauber, M. Wa¨hlisch, C. Deuchler, and R. Wagner (2006), Working models for spatial distribution and level of Mars' seismicity, J. Geophys. Res., 111, E11006, doi:10.1029/2006JE002708.

Mège, D., & Masson P., 1996 Amounts of crustal stretching in Valles Marineris, Mars. Planetary and Space Science, Vol. 44, Issue 8, pp. 749-781.

Minar, J., & Evans, I.S., 2008. Elementary forms for land surface segmentation: The theoretical basis of terrain analysis and geomorphological mapping. Geomorphology, Vol. 95, Issues 3-4, pp. 236-259. doi:10.1016/j.geomorph.2007.06.003

Moellering, H., and A.J. Kimerling, 1990. A New Digital Slope-Aspect Display Process, Cartography and Geographic Information Systems, 17(2): 151-159

Moratto, Z. M., Broxton, M. J., Beyer, R. A., Lundy, M., & Husmann, K., 2010. Ames Stereo Pipeline, NASA's Open Source Automated Stereogrammetry Software. 41st Lunar and Planetary Science Conference, held March 1-5, 2010 in The Woodlands, Texas. LPI Contribution No. 1533, p.2364

Neter, J., & Wasserman, W., 1974. Applied Linear Statistics Models. First Edition. Richard D. Irwin. ISBN 978-0256014983

Patton, P.C., 1990. Erosional development of bedrock spur and gully topography in the Valles Marineris, Mars. Final Report for NASA Grant NAGW-0244. Wesleyan University.

Peulvast, J.-P., D. Mège, J. Chiciak, F. Costard, P. L. Masson, (2001), Morphology, evolution and tectonics of Valles Marineris wallslopes (Mars). Geomorphology 37, 329 – 352.

Roberts, A.W., 1982. Elementary linear algebra. The Benjamin/Cummings Publishing Company, Inc. Menlo Park,, California. ISBN 0-8053-8302-6

Russ, J.C., 1992.The Image Processing Handbook. CRC Press, Inc., Boca Raton, Florida. ISBN 0-8493-4233-3

Schmidt, G., 2015. Geology of Hebes Chasma, Valles Marineris, Mars. M.Sc. Thesis, Brock University

Sclaroff, S., & Pentland, A., 1991. Generalized Implicit Functions For Computer Graphics. Computer Graphics, Vol. 25, No. 4, pp. 247-250.

Sharpnack, D.A. and Akin, G., 1969. An algorithm for computing slope and aspect from elevations. Photogramm. Eng., 35: 247-248.

Skidmore, A.K., 1989. A Comparison of Techniques for Calculating Gradient and Aspect from a Gridded Digital Elevation Model. International Journal of Geographical Information Systems, Vol. 3, No. 4, 323-334

Slocum, T.A., McMaster, R.B., Kessler, F.C., & Howard, H.H., 2009. Thematic Cartography and Geovisualization. Third Edition. Pearson Prentice Hall, Upper Saddle River, NJ. ISBN978-0-13-229834-6

Smith, J., 2013. Measuring Planar Features using CTX Images and Digital Terrain Maps: Searching for Evidence of Faulting in Candor, Ophir and Coprates Chasmata, Valles Marineris, Mars. B.Sc. Thesis, Brock University

Styan, G.P.H., 1973 Hadamard products and multivariate statistical analysis. Linear Algebra and its Applications, vol. 6, 1973, pp. 217-240. doi:10.1016/0024-3795(73)90023-2

Turner, F.J., & Weiss, L.E., 1963. Structural analysis of metamorphic tectonites. McGraw-Hill, New York.

Zevenbergen, L.W., & Thorne, C.R., 1987. Quantitative analysis of land surface topography. Earth surface processes and landforms, Vol. 12, 47-56. John Wiley & Sons, Ltd.

Appendixes

APPENDIX 1. WEB LINKS

Electronic

APPENDIX 2. VISUAL STUDIO VB.NET PROJECT

Electronic

APPENDIX 3. ARCGIS MODELS

Electronic

APPENDIX 4. PYTHON SCRIPTS

Electronic

APPENDIX 5. MATHEMATICAL DERIVATIONS

5.1: Differences between adjacent pixels on a kernel cancel out on summation

By assigning origin to the center of the kernel, calculations for any kernel with x-scaling X, x-radius n, and y-scaling Y, y-radius m, can be simplified as follows:

$$\sum_{i=-n}^{n} x_j = \sum_{j=-n}^{n} jX = X\sum_{j=-n}^{n} j = X\left(\left(\sum_{j=1}^{n} j\right) + 0 + \left(\sum_{j=-n}^{-1} j\right)\right) = X\left(\left(\sum_{j=1}^{n} j\right) + \left(\sum_{j=1}^{n} -j\right)\right) = X\left(\left(\sum_{j=1}^{n} j\right) - \left(\sum_{j=1}^{n} j\right)\right) = 0$$

Similarly, $\sum_{k=-m}^{m} y_k = 0$;

$$\sum_{j=-n,k=-m}^{n,m} x_j\, y_k = \sum_{j=-n,k=-m}^{n,m} XiYj = XY\sum_{j=-n,k=-m}^{n,m} jk = XY\left(\left(\sum_{j=1,k=-m}^{n,m} jk\right) + \left(\sum_{j=-n,k=-m}^{-1,m} jk\right)\right) = XY\left(\left(\sum_{j=1,k=-m}^{n,m} jk\right) + \left(\sum_{j=1,k=-m}^{n,m} -ij\right)\right) = XY\left(\left(\sum_{j=1,k=-m}^{n,m} jk\right) - \left(\sum_{j=1,k=-m}^{n,m} jk\right)\right) = 0$$

5.2: Calculating normalization coefficient for square kernel

Since $\begin{cases} a * \sum_{j=-n,k=-m}^{n,m} {x_{j,k}}^2 = -\sum_{j=-\mathrm{n},k=-\mathrm{m}}^{n,m} z_{j,k}x_{j,k} \\ b * \sum_{j=-n,k=-m}^{n,m} {y_{j,k}}^2 = -\sum_{j=-\mathrm{n},k=-\mathrm{m}}^{n,m} z_{j,k}y_{j,k} \end{cases}$;

Since $\sum_{j=-\mathrm{n},k=-\mathrm{m}}^{n,m} z_{j,k}x_{j,k} = \sum_{j=-\mathrm{n},k=-\mathrm{m}}^{n,m} z_{j,k}Xj = X\sum_{j=-\mathrm{n},k=-\mathrm{m}}^{n,m} z_{j,k}j$, and similarly

$\sum_{j=-\mathrm{n},k=-\mathrm{m}}^{n,m} z_{j,k}y_{j,k} = Y\sum_{j=-\mathrm{n},k=-\mathrm{m}}^{n,m} z_{j,k}k$, and also

$\sum_{j=-n,k=-m}^{n,m} {x_{j,k}}^2 = \sum_{j=-n,k=-m}^{n,m} (Xj)^2 = X^2\sum_{j=-n,k=-m}^{n,m} j^2 =$

$X^2(2m+1)\sum_{j=-n}^{n} j^2 = X^2(2m+1)\left(\left(\sum_{j=-n}^{-1} j^2\right) + 0 + \left(\sum_{j=1}^{n} j^2\right)\right) =$

$X^2(2m+1)\left(\left(\sum_{j=1}^{n} (-j)^2\right) + 0 + \left(\sum_{j=1}^{n} j^2\right)\right) = 2X^2(2m+1)\left(\sum_{j=1}^{n} j^2\right)$, similarly

$\sum_{j=-n,k=-m}^{n,m} {y_{j,k}}^2 = 2Y^2(2n+1)\left(\sum_{k=1}^{m} k^2\right)$.

Meanwhile $\sum_{j=1}^{n} j^2 = \frac{n(n+1)(2n+1)}{6}$ and $\sum_{k=1}^{m} k^2 = \frac{m(m+1)(2m+1)}{6}$,

Therefore,

$a = {-\sum_{j=-\mathrm{n},k=-\mathrm{m}}^{n,m} z_{j,k}x_{j,k}} \Big/ {\sum_{j=-n,k=-m}^{n,m} {x_{j,k}}^2} =$

$-X\sum_{j=-\mathrm{n},k=-\mathrm{m}}^{n,m} z_{i,j}j \Big/ 2X^2(2m+1)\left(\sum_{j=1}^{n} j^2\right) =$

$-\sum_{j=-\mathrm{n},k=-\mathrm{m}}^{n,m} z_{j,k}j \Big/ 2X(2m+1)\left(\frac{n(n+1)(2n+1)}{6}\right) =$

$-\sum_{j=-\mathrm{n},k=-\mathrm{m}}^{n,m} z_{j,k}j \Big/ \left(\frac{n(n+1)(2n+1)(2m+1)}{3}\right) = \frac{-3\sum_{j=-\mathrm{n},k=-\mathrm{m}}^{n,m} z_{j,k}j}{Xn(n+1)(2n+1)(2m+1)}$

.

And likewise, $b = \frac{-3\sum_{j=-\mathrm{n},k=-\mathrm{m}}^{n,m} z_{j,k}k}{Ym(m+1)(2m+1)(2n+1)}$.

5.3.1: Calculating Strike

The coefficient *d* is irrelevant to determining the strike and dip of the fitted plane $ax + by + z + d = 0$, so setting $d = -1$ results in a plane equation $ax + by + z = 1$.

Since strike line is given by the intersection of model plane with *XY* plane, it is defined by points $X\left(\frac{1}{a},0,0\right)$ and $Y\left(0,\frac{1}{b},0\right)$. The strike vector, in conformance with the "American Right Hand Rule", is obtained by translating X by $-\overrightarrow{OY}$, i.e. it is vector $\overrightarrow{YX} = (\frac{1}{a},\frac{-1}{b},0)$, the strike value, consequently, is an angle measured clockwise from $\overrightarrow{OY}$ to $\overrightarrow{YX}$, hence

$$Strike = \tan^{-1}\frac{1/a}{-1/b} = \tan^{-1}\frac{-b}{a} =$$

$$\tan^{-1}\left(\frac{-(-3\,(Z*K))}{Ym(m+1)(2m+1)(2n+1)} \Big/ \frac{-3\,(Z*J)}{Xn(n+1)(2n+1)(2m+1)}\right) =$$

$$\tan^{-1}\left(\frac{-3\,(Z*K)}{3\,(Z*J)} \Big/ \frac{Ym(m+1)(2m+1)(2n+1)}{Xn(n+1)(2n+1)(2m+1)}\right) = \tan^{-1}\left(\frac{-(Z*K)}{(Z*J)} \Big/ \frac{Ym(m+1)}{Xn(n+1)}\right) =$$

$$\tan^{-1}\left(-\left(\frac{Xn(n+1)}{Ym(m+1)}\right)\left(\frac{(Z*K)}{(Z*J)}\right)\right) = \tan^{-1}\left(\left(\frac{Xn(n+1)}{Ym(m+1)}\right)\left(\frac{(Z*(-K))}{(Z*J)}\right)\right)$$

For a square kernel on a simple equispaced raster, $Xn(n+1) = Ym(m+1)$, therefore further simplification is possible, and $Strike = \tan^{-1}\frac{(Z*(-K))}{(Z*J)}$.

5.3.2: Calculating Dip

Since the coefficients *c=1* and *d=-1*, at x=0, y=0 z=1, so the dip of the plane can be found as

$$Dip = \tan^{-1}\frac{1}{distance\ from\ origin\ to\ segment\ XY}$$

On a horizontal plane at a datum elevation, i.e. z=0, the equation of a strike line is ax+by+cz+d=0 => ax+by-1=0, hence distance from origin to the segment XY is $\frac{1}{\sqrt{a^2+b^2}}$, therefore

$$Dip = \tan^{-1}\frac{1}{1/\sqrt{a^2+b^2}} = \tan^{-1}\sqrt{a^2+b^2} =$$

$$\tan^{-1}\sqrt{\left(\frac{-3\,(Z*J)}{Xn(n+1)(2n+1)(2m+1)}\right)^2+\left(\frac{-3\,(Z*K)}{Ym(m+1)(2m+1)(2n+1)}\right)^2} =$$

$$\tan^{-1}\sqrt{\left(\frac{3}{(2n+1)(2m+1)}\right)^2\left(\left(\frac{(Z*J)}{Xn(n+1)}\right)^2+\left(\frac{(Z*K)}{Ym(m+1)}\right)^2\right)} =$$

$$\tan^{-1}\left(\left(\frac{3}{(2n+1)(2m+1)}\right)\sqrt{\frac{(Z*J)^2}{(Xn(n+1))^2}+\frac{(Z*K)^2}{(Ym(m+1))^2}}\right) =$$

$$\tan^{-1}\left(\left(\frac{3}{(2n+1)(2m+1)}\right)\sqrt{\frac{(Z*J)^2(Ym(m+1))^2+(Z*K)^2(Xn(n+1))^2}{(Xn(n+1)Ym(m+1))^2}}\right) =$$

$$\tan^{-1}\left(\frac{3\sqrt{(Z*J)^2(Ym(m+1))^2+(Z*K)^2(Xn(n+1))^2}}{(2n+1)(2m+1)Xn(n+1)Ym(m+1)}\right)$$

For a square kernel on an equispaced raster, this simplifies to
$Dip = \tan^{-1}\left(\frac{3\sqrt{(Z*J)^2+(Z*K)^2}}{(2n+1)(2m+1)}\right)$.

5.4: Calculating Sum of Squared Errors as a second derivative

Let Observed = T; Predicted = P, raster source data = D, gradient along x-axis = X, gradient along y-axis = Y, distance from neighbourhood centre along x-axis = i, same along y = j, then:

$$\begin{aligned}\sum(T-P)^2 &= \sum(T^2-2TP+P^2)\\ &= \left(\sum D^2\right)-\sum\left(2D\left(\frac{\sum D}{N}+Xi+Yj\right)\right)+\sum\left(\left(\frac{\sum D}{N}+Xi+Yj\right)^2\right)\\ &= \left(\sum D^2\right)-\frac{2}{N}\sum\left(D\sum D\right)-2X\sum Di-2Y\sum Dj\\ &\quad+\sum\left(\left(\frac{\sum D}{N}+Xi+Yj\right)^2\right)\end{aligned}$$

Since $\sum Di = X\sum i^2$, *and* $\sum Dj = Y\sum j^2$, *above can be simplified to*

$$\sum (T-P)^2 = \left(\sum D^2\right) - \frac{2}{N}\sum\left(D\sum D\right) - 2X^2\sum i^2 - 2Y^2\sum j^2 + \sum\left(\left(\frac{\sum D}{N} + Xi + Yj\right)^2\right)$$

$$= \left(\sum D^2\right) - \frac{2}{N}\sum\left(D\sum D\right) - 2\left(X^2\sum i^2 + Y^2\sum j^2\right) + \sum\left(\left(\frac{\sum D}{N}\right)^2 + 2\frac{\sum D}{N}(Xi + Yj) + (Xi + Yj)^2\right)$$

$$= \left(\sum D^2\right) - \frac{2}{N}\sum\left(D\sum D\right) - 2\left(X^2\sum i^2 + Y^2\sum j^2\right) + \sum\left(\frac{\sum D}{N}\right)^2 + \frac{2X\sum D}{N}\sum(i) + \frac{2X\sum D}{N}\sum(j) + \sum(X^2i^2) + 2XY\sum ij + \sum(Y^2j^2)$$

Since $\sum(j) = 0,\ \sum(i) = 0, \sum(D\sum D) = (\sum D)^2$, this can be simplified to

$$\sum (T-P)^2 = \left(\sum D^2\right) - \frac{2}{N}\left(\sum D\right)^2 - 2\left(X^2\sum i^2 + Y^2\sum j^2\right) + \sum\left(\frac{\sum D}{N}\right)^2 + \sum(X^2i^2) + \sum(Y^2j^2)$$

$$= \left(\sum D^2\right) - \frac{2}{N}\left(\sum D\right)^2 - 2\left(X^2\sum i^2 + Y^2\sum j^2\right) + \left(\frac{\sum D}{N}\right)^2 N + \sum(X^2i^2) + \sum(Y^2j^2)$$

$$= \left(\sum D^2\right) - \frac{2(\sum D)^2}{N} - 2\left(\left(X^2\sum i^2 + Y^2\sum j^2\right)\right) + \frac{(\sum D)^2}{N} + \sum(X^2i^2) + \sum(Y^2j^2)$$

$$= \left(\sum D^2\right) - 2\left(\left(X^2\sum i^2 + Y^2\sum j^2\right)\right) - \frac{(\sum D)^2}{N} + X^2\sum(i^2) + Y^2\sum(j^2) = \sum(D^2) - \frac{(\sum D)^2}{N} - X^2\sum(i^2) - Y^2\sum(j^2)$$

For a kernel that is (2m+1) pixels tall and (2n+1) pixels wide,

$$\sum(i^2) = \frac{n(n+1)(2n+1)(2m+1)}{3};\ \sum(j^2) = \frac{m(m+1)(2m+1)(2n+1)}{3}$$

reduces to

$$\sum (T-P)^2 = \left(\sum D^2\right) - \frac{(\sum D)^2}{N} - X^2n(n+1)(2n+1)(2m+1)/3 - Y^2m(m+1)(2m+1)(2n+1)/3$$

$$= \left(\sum D^2\right) - \frac{(\sum D)^2}{N} - \frac{X^2n(n+1)N}{3} - \frac{Y^2m(m+1)N}{3}$$